# 作业

## 22920212204392 黄勖

3.7假设 myword 是一个字变量，mybyte1 和 mybyte2 是两个字节变量，指出下列语句中的错误原因。

（1） mov byte ptr [bx],1000

（2） mov bx,offset myword[si]

（3） cmp mybyte1,mybyte2

（4） mov al,mybyte1+mybyte2

（5） sub al,myword

（6） jnz myword

答：

（1）执行mov byte ptr [bx],1000时1000 超出了一个字节范围数据类型不匹配，目的地址单元是字节单元，1000大于255。无法存放。

（2）执行mov bx,offset myword[si] 时offset是汇编过程计算的偏移地址，故offset后只能跟标识符。寄存器的值只有程序执行时才能确定，故此处无法确定，可以改为 lea bx,myword[si]

（3）执行cmp mybyte1,mybyte2时两个都是存储单元，指令不允许，即CMP的两个操作数不能同时在存储器中。

（4）执行mov al,mybyte1+mybyte2时汇编过程不能计算（变量值只有执行时才确定）即运算符只能对常数进行操作，不能对变量进行操作。

（5）执行sub al,myword时数据类型不匹配，al是8位的，myword是16位变量。此处同时使用了字节量 AL 与字量 myword，二者类型不匹配

（6）执行jnz myword 时myword是变量，其指的单元中存放的是操作数，不能用做跳转入口。注：Jcc 指令只有相对寻址方式，不支持间接寻址方式

3.10画图说明下列语句分配的存储空间及初始化的数据值：

（1） byte\_var DB ‘ABC’,10,10h,‘EF’,3 DUP( -1,?,3 DUP(4))

（2） word\_var DW 10h,- 5,‘EF’,3 DUP(?)

答：

1. 从低地址开始，依次是（十六进制表达）：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ‘A’ | ‘B’ | ‘C’ | 10 | 10H | ‘e’ | ‘f’ | -1 | — | 4 | 4 | 4 | … |
| 41h | 42h | 43h | 0ah | 10h | 45h | 46h | ffh | ? | 04h | 04h | 04h |  |

1. 从低地址开始，依次是（十六进制表达）：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10H | 0 | FBH | FFH | 46H | 45H | — | — | — | — | …… |
| 10h | | -5 | | ‘EF’ | | ？ | | ？ | |  |

3.18在 SMALL 存储模式下，简化段定义格式的代码段、数据段和堆栈段的缺省段名、定位、组合以及类别属性分别是什么？

答：

段定义伪指令 段名 定位 组合 类别 组名

.CODE \_TEXT WORD PUBLIC 'CODE'

.DATA \_DATA WORD PUBLIC 'DATA' DGROUP

.DATA？ \_BSS WORD PUBLIC 'BSS' DGROUP

.STACK STACK PARA STACK 'STACK' DGROUP

3.21按下面要求写一个简化段定义格式的源程序

（1） 定义常量 num ，其值为5 ；数据段中定义字数组变量 datalist ，它的前 5 个字单元中依次存放 -1 、 0 、2 、5 和 4 ，最后 1 个单元初值不定；

（2） 将代码段中的程序将 datalist 中头前num 个数的累加和存入 datalist 的最后 1 个字单元中。

答：

.model small

.stack

.data

num equ 5

datalist dw -1,0,2,5,4,?

.code

.startup

mov bx,offset datalist

mov cx,num

xor ax,ax

again: add ax,[bx]

inc bx

inc bx ; 因为datalist是字类型数组，每个元素占用两个字节，inc一次只能跳到当前元素高位字节，两次可确保bx指向下一个元素

loop again

mov [bx],ax

.exit 0

end

注：①dw 字值(2 个字节)。每个显示行显示行中第一个单词的地址和最多八个十六进制单词值。默认计数为64字(128 字节) 。

②ADD AX, [BX]这句代码实际上是双字节加法，例如MOV AX, [BX]把4000H和4001H处的值放入AX，ADD BX, 1后BX = 4001H，之后再把4001H和4002H处的值加在AX上，取了两个字节所以INC两次。