

## 1. 段地址:偏移地址

编程的时候只能使用逻辑地址，不能使用物理地址。

段起始地址的 16 进制个位必须=0，否则不能成为段起始地址

段长度=2233F-12340+1=22340-12340=10000h=64K

1234 段 1235:0048	段起始地址	12340h	00h	1234:0000
		12341h	11h	1234:0001
		.....	22h	
		12350h	33h	段起始地址
		.....	44h	
		12398h	55h	1234:0058
		.....	66h	1235 段
	段结束地址	2233Fh	77h	1234:FFFF
		.....	88h	
		2234Fh	99h	段结束地址

一个物理地址可以表示成多个逻辑地址，例如：

12398h=1234:0058=1235:0048=1236:0038

=1230:0098

段地址的 1 相当于偏移地址的 10h：

段地址+1 就意味着偏移地址+10h；

段地址-1 就意味着偏移地址-10h；

## 2. 直接寻址、间接寻址

设从地址 1000h:2000h 起存放以下 4 个字节：

1000:2000 12h

1000:2001 34h

1000:2002 56h

1000:2003 78h

现编程计算这 4 个字节之和并保存到寄存器 AL 中：

```

mov ax, 1000h
mov ds, ax
mov al, 0
mov bx, 2000h
mov cx, 4
next:
add al, ds:[bx]
add bx, 1
sub cx, 1
jnz next

```

```

mov ax, 1000h
mov ds, ax; ds 是数据段寄存器
除了 cs 不能通过 mov 修改外, ds、es、ss 均可以修改。

```

假定要取出 1000:2000 指向的字节并保存到 AL 中, 则有两种方法:

(1) 直接寻址(用常数来表示变量的偏移地址)

```

mov al, 1000h:[2000h]; 语法错误: 段地址必须用段
                        ; 寄存器来表示, 不能用常数
mov al, ds:[2000h]; ds:2000 可以看作一个指针
                        ; ds:[2000h]表示上述指针指向
                        ; 的对象
mov al, byte ptr ds:[2000h]
                        ; byteptr 表示对象的类型是 byte
                        ; ptr 是 pointer 的缩写

```

可以把上述语句转化成 C 语言语法来写:

```

typedef unsigned char byte;

```

```
al = *(byte *) (ds:2000h) ;
```

(2) 间接寻址 (用寄存器、寄存器+常数来表示变量的偏移地址)

```
mov bx, 2000h
```

```
mov al, ds:[bx] ; 用于间接寻址的寄存器仅限于:
```

```
; bx, bp, si, di
```

```
mov al, byte ptr ds:[bx]
```

什么样的情况下, 必须使用

byte ptr、word ptr、dword ptr

这三种类型修饰?

```
mov ds:[bx], 1; 语法错误
```

```
mov byte ptr ds:[bx], 1; 正确①
```

```
mov word ptr ds:[bx], 1; 正确②
```

```
mov dword ptr ds:[bx], 1; 正确③
```

```
1000:2000 12h → 01h → 01h → 01h
```

```
1000:2001 34h ① → 00h → 00h
```

```
1000:2002 56h ② → 00h
```

```
1000:2003 78h → 00h
```

③

### 3. 如何引用数组元素

引用数组元素的例子: <http://10.71.45.100/bhh/addr.asm>

```
mov ah, [abc] 中,
```

[abc]可以理解成地址 abc 所指向的对象。

mov ah, [abc]的完整形式其实是:

```
mov ah, byte ptr ds:[abc]
```

其中 ds 是变量 abc 的段地址

byte ptr 表示地址 abc 所指的对象是一个字节。

若[]中不包含寄存器 bp, 则该变量默认的段地址一定是 ds, 故在源程序 addr.asm 中可以省略 ds:。

byte ptr 有点类似于 C 语言中的(char \*), 其中 ptr 是单词 pointer 的缩写。

`byte ptr ds:[abc]` 表示地址 `ds:abc` 所指的对象是一个 `byte`。相当于 C 语言的如下描述：

`*(char *) (ds:abc)`

汇编语言的语句中，如果源操作数或目标操作数的其中之一有明确的类型即宽度，则另外一方不需要指定类型。

在本例中，由于 `ah` 是 8 位宽度，故可以省略源操作数的类型 `byte ptr`。

如何在 `dosbox86` 中调试程序：

1. 先编译连接
2. `debug->debug`
3. `view->cpu` 查看机器语言
4. `F8` 单步跟踪

如何在命令行中调试：

`masm addr;`

`link addr;`

`td addr`

其中 `td` 指 Turbo Debugger。

微软对应的调试器叫 `CodeView`，但功能不如 `td`。

C 语言中引用数组元素的例子：<http://10.71.45.100/bhh/addr.c>

编译调试 `addr.c` 的步骤：

复制 `addr.c` 到 `dosbox86\tc` 中

`dosbox86->file->DOS shell`

`cd \tc`

`tc`

按 `Alt+F` 选择菜单 `file->load->addr.c`

按 `Alt+C` 选择菜单 `compile->build all`

按 `Alt+F` 选择菜单 `file->quit`

`td addr`

`view->cpu`

按 `F5` 放大窗口

`F8` 单步调试

调试完成按 `file->quit`

`exit` 返回 `dosbox86` 集成环境

输入一个字符串再输出的例子：<http://10.71.45.100/bhh/io.asm>

用汇编语言控制文本方式下整个屏幕的输出：

屏幕坐标示意图：<http://10.71.45.100/bhh/screen.bmp>

输出 2000 个 'A' 的例子：<http://10.71.45.100/bhh/2000a.asm>

屏幕上移动的 'A' 的例子：<http://10.71.45.100/bhh/mova.asm>