

王爽汇编第七章,更灵活的定位内存地址的方法

1. and 和 or指令
2. 关于ASCII码
3. 显示hello,world!
4. 大小写转换
5. [bx+idata]寻址
6. si和di
7. 不同的寻址方式的灵活应用

1. and 和 or指令

首先, 介绍两条指令 **and** 和 **or**, 因为我们下面的例程中要用到它们。

(1) **and** 指令: 逻辑与指令, 按位进行与运算。

例如指令:

```
mov al,01100011B
and al,00111011B
```

0	1	1	0	0	1	1
0	0	1	1	1	0	1
<hr/>						
0	0	1	0	0	0	1

执行后: al=00100011B

and: 全是1才为1

通过该指令可将操作对象的相应位设为 0, 其他位不变。

例如:

将 al 的第 6 位设为 0 的指令是: **and al,10111111B**

将 al 的第 7 位设为 0 的指令是: **and al,01111111B**

将 al 的第 0 位设为 0 的指令是: **and al,11111110B**

(2) **or** 指令: 逻辑或指令, 按位进行或运算。

例如指令:

```
mov al,01100011B
or al,00111011B
```

0	1	1	0	0	1	1
0	0	1	1	1	0	1
<hr/>						
0	1	1	1	1	0	1

执行后: al=01111011B

or: 有1说1

通过该指令可将操作对象的相应位设为 1, 其他位不变。

例如:

将 al 的第 6 位设为 1 的指令是: **or al,01000000B**

将 al 的第 7 位设为 1 的指令是: **or al,10000000B**

将 al 的第 0 位设为 1 的指令是: **or al,00000001B**

2. 关于ASCII码

ASCII 由电报码发展而来。第一版标准发布于1963年[3] [4] , 1967年经历了一次主要修订[5] [6] , 最后一次更新则是在1986年, 至今为止共定义了128个 **字符**; 其中33个字符无法显示 (一些终端提供了扩展, 使得这些字符可显示为诸如笑脸、**扑克牌** 花式等8-bit符号), 且这33个字符多数都已是陈废的 **控制字符**。控制字符的用途主要是用来操控已经处理过的文字。在33个字符之外的是95个可显示的字符。用 **键盘** 敲下空白键所产生的空白字符也算1个可显示字符 (显示为空白)。

注：表中的ASCII字符可以用：ALT + “小键盘上的数字键”输入

注意这里我调用了 `int 21h`，这是一个DOS的中断，其中ah=09是他的功能号用来打印字符串。

```

1  assume cs:code,ds:data
2  ;>>>>>>>>>>>>>>>
3  ;数据段
4  ;>>>>>>>>>>>>>>>
5  data segment
6      db 'hello,world!','$'
7  data ends
8
9  ;>>>>>>>>>>>>>>>
10 ;代码段
11 ;>>>>>>>>>>>>>>>
12 code segment
13 start:
14     mov ax,data
15     mov ds,ax
16
17     ;调用DOS中断显示字符串
18     mov dx,0 ;ds:[dx]
19     mov ah,09h
20     int 21h

```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
D:\>T.EXE
hello,world!
D:\>
```

小写字母的ASCII码值比大写字母的ASCII码值大20H(32)
大写字母ASCII码的第5位为0, 小写字母的第5位为1(其他一致)

大写	十六进制	二进制	小写	十六进制	二进制
A	41	01000001	a	61	01100001
B	42	01000010	b	62	01100010
C	43	01000011	c	63	01100011
D	44	01000100	d	64	01100100
E	45	01000101	e	65	01100101
F	46	01000110	f	66	01100110

```

1 ;*****
2 ;* *
3 ;* 要求：将所有字符串转换成小写字符串 *
4 ;*****
5
6 assume cs:code,ds:data
7 ;>>>>>>>>>>>>
8 ;数据段
9 ;>>>>>>>>>>>>
10 data segment
11     db 'Hello,World!','$'
12 data ends
13 ;>>>>>>>>>>>>
14 ;代码段
15 ;>>>>>>>>>>>>
16 code segment
17 main:
18     mov ax,data ;ds段
19     mov ds,ax
20
21     ;大小写转换
22     mov bx,0
23     mov cx,11
24 s: mov al,ds:[bx] ;将ASCII码从 ds:bx单元中取出
25     or al,00100000b;让第5位无论如何都变1

```

```

26      mov ds:[bx],al
27      inc bx
28      loop s
29
30      ;输出字符串
31      mov dx,0; ds:[0]
32      mov ah,09h
33      int 21h
34
35      ;retun 2 dos
36      mov ax,4c00h
37      int 21h
38 code ends
39 end main

```

```
7 ;>>>>>>>>>>>>>>>>
8 ;数据段
9 >>>>>>>>>>>>>>>>
10 data segment
11     db 'Hello,World!','$'
```

DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program:

```
D:\>AZADEMO.EXE
hello,world!
```

D:\>

成功将首字母大写替换成了小写

```
D:\>AZADEMO.EXE
hello,world!
D:\>_
```

```
7 ;>>>>>>>>>>>>>>>
8 ;数据段
9 >>>>>>>>>>>>>>>
10 data segment
11     db 'Hello,World!','$'
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program:

```
D:\>AZADEMO.EXE
hello,world!
```

D:\>

成功将首字母大写替换成了小写

5. [bx+idata]寻址

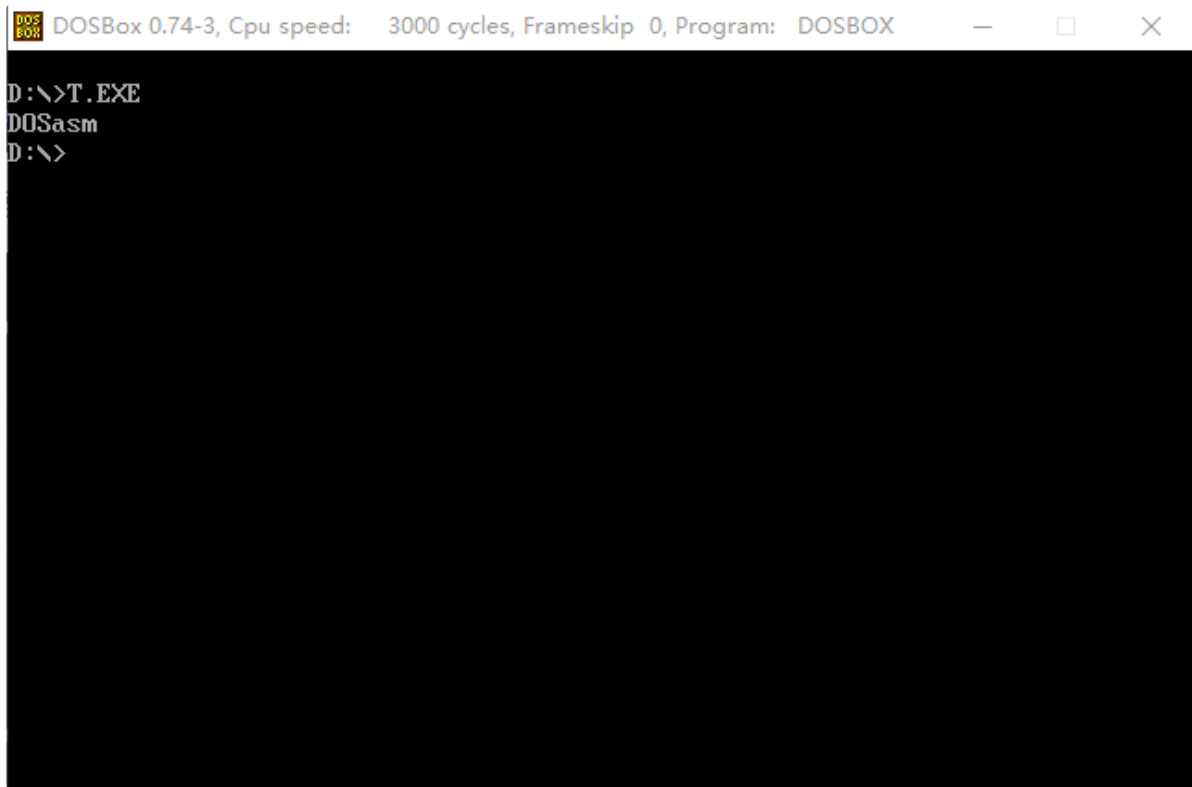
在前面，我们用[bx]的方式来指明一个内存单元，还可以用一种更为灵活的方式来指明内存单元：[bx+idata]表示一个内存单元，他的偏移地址为(bx)+idata [bx中的数值加上数字]

例如：

```
mov ax,[bx+200]
```

也可以写成如下的格式：（注意这里指的是masm所支持的伪指令，具体其他编译器看情况而定。）

[bx+idata]处理数组例子:



6. si和di

这两个寄存器其实之前介绍寄存器篇章的时候，已经介绍过了，si和di都是偏移寄存器并且注意在x86架构下，他们是字符串相关的寄存器。

```
(1) mov bx,0  
    mov ax,[bx]
```

```
(2) mov si,0  
    mov ax,[si]
```

```
(3) mov di,0  
    mov ax,[di]
```

下面的3组指令也实现了相同的功能。

```
(1) mov bx,0  
    mov ax,[bx+123]
```

```
(2) mov si,0  
    mov ax,[si+123]
```

```
(3) mov di,0  
    mov ax,[di+123]
```

7. 不同的寻址方式的灵活应用

如果我们比较一下前面用到的集中定位内存地址的方法(可以称为寻址方式)，就可以发现：

- [idata]用一个常量来表示地址，可用于直接定位一个内存单元；
- [bx]用一个变量来表示地址，可用于间接定位一个内存单元(比如循环)；

- `[bx+idata]`用一个变量和常量表示地址，可在一个起始地址的基础上用变量间接定位一个内存单元(比如多个数组);
- `[bx+si]`用两个变量表示地址;
- `[bx+si+idata]`用连个变量和一个常量表示地址。

可以看到，从`[idata]`一直到`[bx+si+idata]`，我们可以用更加灵活的方式来定位一个内存单元的地址。这使我们可以从更结构化的角度 来对待所要处理的数据。

参考文献：

<https://blog.csdn.net/trochiluses/article/details/20008631> int 21—dos系统调用 表

https://blog.csdn.net/qq_39654127/article/details/88698911 王爽汇编笔记

《王爽汇编第4版》 第7章