

# 《汇编语言》王爽 P88 实验3

原创

酱油瓶被人注册了  于 2018-03-07 14:38:26 发布  5009  收藏 16

分类专栏: [汇编](#) 文章标签: [汇编](#) [王爽](#) [实验3](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_25175067/article/details/79470838](https://blog.csdn.net/qq_25175067/article/details/79470838)

版权



[汇编](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

## 《汇编语言》王爽 P88 实验3

Mac系统配置这些蛮麻烦的:

下载个DOSBOX, 再下载debug, 在DOSBOX中把debug挂到C盘去吧

比如我想把air下面的debug文件挂成C盘, 就括号中这条命令:

(mount c /Users/air/debug), 当然你也可以放其他位置, 取其他的名字, 只要后面的目录对就可以;

下载edit.exe 和masm都放到debug这个文件中去吧, 这样执行起来也方便

1) 生成t1.exe (注: 数字后面一定要加h, 我一开始很蠢的把)

```
assume cs : codeseg
codeseg segment
    mov ax, 2000H
    mov ss, ax
    mov sp, 0
    add sp, 4
    pop ax
    pop bx
    push ax
    push bx
    pop ax
    pop bx
    mov ax, 4c00H
    int 21h
codeseg ends
end
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Source filename [.ASM]: T1
Object filename [T1.OBJ]: T1
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51806 + 464738 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\MASM>link

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Object Modules [.OBJ]: T1
Run File [T1.EXE]: T1
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>_
```

[http://blog.csdn.net/qq\\_25175067](http://blog.csdn.net/qq_25175067)

2) debug执行T1.exe之后:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c /Users/air/debug
Drive C is mounted as local directory /Users/air/debug/

Z:\>c:

C:\>debug T1.exe
-r
AX=FFFF BX=0000 CX=0016 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000 NU UP EI PL NZ NA PO NC
076A:0000 B80020 MDU AX,2000
```

[http://blog.csdn.net/qq\\_25175067](http://blog.csdn.net/qq_25175067)

可以看出 DS = 075A; CS=076A, 两个段地址相差10H, 是因为PSP的256个字节的原因;

疑惑: DS和CS的不同, 有什么用嘛?

CX=0016是T1的字节长度, 16H个字节;

从mov ax,2000到int 21的每个指令的字节长度为: 3, 2, 3, 3, 1, 1, 1, 1, 1, 1, 3, 2, sum之后是22个字节, 写成16进制就是16H(16\*1+6=22).

以下为执行mov ax, 2000H

mov ss, ax

mov sp, 0

add sp, 4

很容易看出来 mov ss, ax 和 mov sp, 0000同时执行了

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
C:\>debug T1.exe
-r
AX=FFFF BX=0000 CX=0016 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000  NU UP EI PL NZ NA PO NC
076A:0000 B80020          MOV     AX,2000
-u
076A:0000 B80020          MOV     AX,2000
076A:0003 8ED0          MOV     SS,AX
076A:0005 BC0000          MOV     SP,0000
076A:0008 83C404          ADD     SP,+04
076A:000B 5B            POP     AX
076A:000C 5B            POP     BX
076A:000D 50            PUSH    AX
076A:000E 53            PUSH    BX
076A:000F 5B            POP     AX
076A:0010 5B            POP     BX
076A:0011 B8004C          MOV     AX,4C00
076A:0014 CD21          INT     21
076A:0016 0000          ADD     [BX+SI],AL
076A:0018 0000          ADD     [BX+SI],AL
076A:001A 0000          ADD     [BX+SI],AL
076A:001C 0000          ADD     [BX+SI],AL
076A:001E 0000          ADD     [BX+SI],AL
_
```

[http://blog.csdn.net/qq\\_25175067](http://blog.csdn.net/qq_25175067)

以下为执行pop ax

pop bx

push ax

push bx

很容易看出来 每次pop之后 SP+2; 每次push之后, SP-2, 初始为SP=0004

pop ax, 把栈顶内容存入寄存器ax, 所以ax从2000编程0000, SP=0006

pop bx, 把栈顶内容存入寄存器bx, SP=0008

push ax, 把寄存器ax内容压入栈顶, SP=0006

猜测一下: push bx, 把寄存器bx内容压入栈顶, SP=0004

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=2000 BX=0000 CX=0016 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=0008  NU UP EI PL NZ NA PO NC
076A:0008 83C404          ADD     SP,+04
-t
AX=2000 BX=0000 CX=0016 DX=0000 SP=0004 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=000B  NU UP EI PL NZ NA PO NC
076A:000B 5B          POP     AX
-t
AX=0000 BX=0000 CX=0016 DX=0000 SP=0006 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=000C  NU UP EI PL NZ NA PO NC
076A:000C 5B          POP     BX
-t
AX=0000 BX=0000 CX=0016 DX=0000 SP=0008 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=000D  NU UP EI PL NZ NA PO NC
076A:000D 50          PUSH    AX
-t
AX=0000 BX=0000 CX=0016 DX=0000 SP=0006 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=000E  NU UP EI PL NZ NA PO NC
076A:000E 53          PUSH    BX
-
http://blog.csdn.net/qq_25175067
```

以下为执行pop ax

pop bx

mov ax,4c00H

int 21h

最后program terminated normally, 一切正常。

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
076A:000E 53          PUSH    BX
-t
AX=0000  BX=0000  CX=0016  DX=0000  SP=0004  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=2000  CS=076A  IP=000F  NU UP EI PL NZ NA PO NC
076A:000F 5B          POP     AX
-t
AX=0000  BX=0000  CX=0016  DX=0000  SP=0006  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=2000  CS=076A  IP=0010  NU UP EI PL NZ NA PO NC
076A:0010 5B          POP     BX
-t
AX=0000  BX=0000  CX=0016  DX=0000  SP=0008  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=2000  CS=076A  IP=0011  NU UP EI PL NZ NA PO NC
076A:0011 B8004C     MOV     AX,4C00
-t
AX=4C00  BX=0000  CX=0016  DX=0000  SP=0008  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=2000  CS=076A  IP=0014  NU UP EI PL NZ NA PO NC
076A:0014 CD21     INT     21
-p
Program terminated normally
http://blog.csdn.net/qq_25175067
```

3) PSP的头两个字节是CD20, 用debug加载t1.exe, 查看PSP的内容

也就是说要查看内存075A:0的内容嘛;

使用命令: -d ds:0, 答案bingo!

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
DS=075A ES=075A SS=2000 CS=076A IP=0010  NU UP EI PL NZ NA PO NC
076A:0010 5B          POP      BX
-t
AX=0000 BX=0000 CX=0016 DX=0000 SP=0008 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=0011  NU UP EI PL NZ NA PO NC
076A:0011 B8004C      MOV     AX,4C00
-t
AX=4C00 BX=0000 CX=0016 DX=0000 SP=0008 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=2000 CS=076A IP=0014  NU UP EI PL NZ NA PO NC
076A:0014 CD21          INT     21
-p
Program terminated normally
-d ds:0
075A:0000 CD 20 FF 9F 00 EA FF FF-AD DE 4F 03 A3 01 8A 03 . . . . .0. . . . .
075A:0010 A3 01 17 03 A3 01 92 01-FF FF FF FF FF FF FF FF . . . . .
075A:0020 FF FF FF FF FF FF FF-FF FF FF FF 50 07 F0 FF . . . . .P. . . . .
075A:0030 00 20 14 00 18 00 5A 07-FF FF FF FF 00 00 00 00 . . . . .Z. . . . .
075A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 . . . . .
075A:0050 CD 21 CB 00 00 00 00 00 00-00 00 00 00 00 00 00 .!. . . . .
075A:0060 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 . . . . .
075A:0070 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 . . . . .
http://blog.csdn.net/qq_25175067
```