

汇编语言开发环境

学习8086CPU汇编语言需要在32位系统DOS环境中进行调试，目前的操作系统一般都是64位且升级到Windows10，最近我也在无数次系统提醒之下按下了Window11更新，这些都无法启动debug.exe。



因此，我们可以使用DOSBox工具，可以在多种操作系统中支持DOS环境，**首先下载好DOSBox和debug.exe，我将这两个可执行文件放到桌面文件夹asm中。**



双击DOSBox安装包，一路下一步流畅无脑安装。然后双击启动：



挂载目录：

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, F...
Welcome to DOSBox v0.74-3
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: c:/Users/Cat/Desktop/asm
Drive C is mounted as local directory c:/Users/Cat/Desktop/asm\
Z:\>c:
C:\>debug
-aa_
```

然而，每次启动都要执行命令，比较麻烦，可以直接找到下面文件：

```
DOSBox Status Window
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox Team, published under GNU GPL.
CONFIG:Loading primary settings from config file C:\Users\Cat\AppData\Local\DOSBox\dosbox-0.74-3.conf
MIDI:Opened device:win32
```

```
1 文件位置（需要设置显示隐藏文件）：
2 C:\Users\Cat\AppData\Local\DOSBox\C:\Users\Cat\AppData\Local\DOSBox
3 在最后面加上指令：
4
5 [autoexec]
6 # Lines in this section will be run at startup.
7 # You can put your MOUNT lines here.
8 mount c: c:/Users/Cat/Desktop/asm
9 c:
```

调整界面大小：

```
1 # windowresolution=original
2 # output=surface
3 windowresolution=1080x800
4 output=opengl
```

入栈出栈

编程：

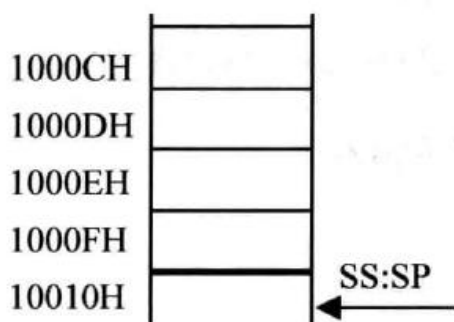
1. 将1000H~1000FH这段空间当作栈，初始状态栈是空的；
2. 设置AX=001AH，BX=001BH；
3. 将AX、BX中的数据入栈；

4. 然后将AX、BX清零；
5. 从栈中恢复AX、BX原来内容。

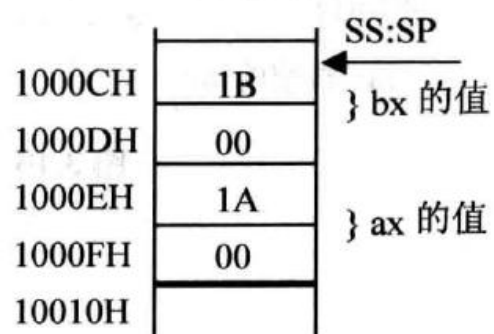
```

-a
073F:0100 mov ax,1000
073F:0103 mov ss,ax
073F:0105 mov sp,0010
073F:0108 mov ax,001A
073F:010B mov bx,001B
073F:010E push ax
073F:010F push bx
073F:0110 sub ax,ax
073F:0112 sub bx,bx
073F:0114 pop bx
073F:0115 pop ax
073F:0116

```



(a) 栈空的情况



(b) ax、bx 入栈的情况

栈交换数据

编程：

1. 将1000H~1000FH这段空间当作栈，初始状态栈是空的；
2. 设置AX=001AH，BX=001BH；
3. 利用栈，交换AX和BX中的数据。

```
1 .....  
2 pop ax  
3 pop bx
```

栈写入数据

在10000H处写入字型数据2266H，可以使用如下程序实现：

```
-a  
073F:0113 mov ax,1000  
073F:0116 mov ds,ax  
073F:0118 mov ax,2266  
073F:011B mov [0],ax  
073F:011E  
  
-d 1000:0 f  
1000:0000 66 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

下面，要求不使用mov 内存单元,寄存器指令实现此功能。

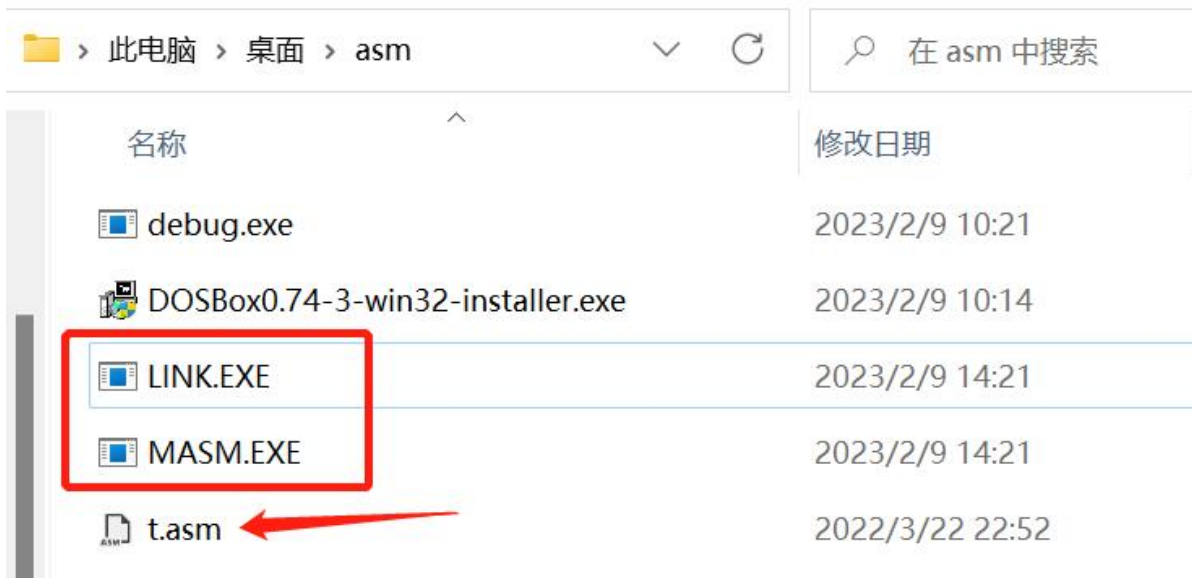
思考，利用入栈操作，设置栈段和栈顶指针再将数据入栈。

```
-a  
073F:011E mov ax,1000  
073F:0121 mov ss,ax  
073F:0123 mov sp,2  
073F:0126 mov ax,2266  
073F:0129 push ax  
073F:012A
```

DOSBox运行asm文件

公众号：黑猫编程

将masm.exe和link.exe放入asm文件夹，并创建t.asm文件，写入程序：网址：<https://noi.hioier.co>



```
C:\>masm t;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51818 + 464726 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link t;
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment
```

如果觉得编译链接过程比较繁琐，可以在配置文件中直接写好，每次执行文件重启DOSBox就可以，可以把DOSBox固定到桌面下方任务栏，文件名指定，比如t.asm，需要测试其他文件就要将新的程序复制过来。

内存数据累加

计算ffff:0~ffff:b单元中的数据的和，结果存储在dx中。

注意：

- 1.单元中数据为8位，不能直接累加到16位寄存器中；
- 2.可以使用一个16位寄存器做中介，将内存单元中的8位数据赋值到一个16位寄存器ax中，再累加到dx上。

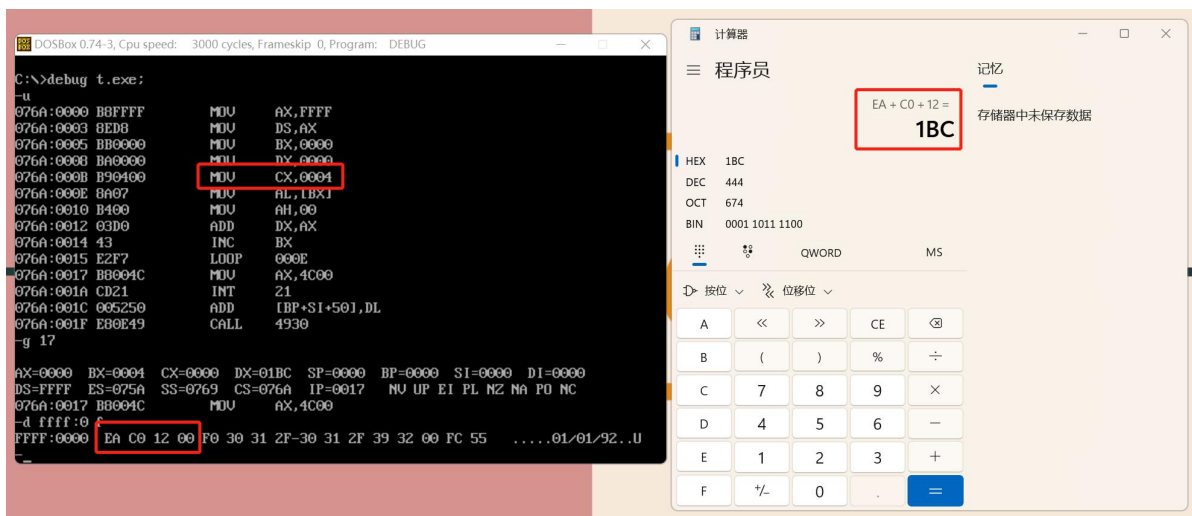
```
1  assume cs:codesg
2
3  codesg segment
4
5  start:mov ax,0ffffH
6         mov ds,ax
7         mov bx,0
```

公众号：黑猫编程
网址：<https://noi.hioqier.co>

```

8
9      mov dx,0
10     mov cx,12
11
12     s:mov al,[bx]
13     mov ah,0
14     add dx,ax
15     inc bx
16     loop s
17
18     mov ax,4c00H
19     int 21H
20
21 code seg ends
22
23 end start

```



测试时使用4个单元数据累加，用计算器进行验证，最终DX寄存器中的数值也是1BCH。