

汇编语言：汇编程序的hello world程序（实验9根据材料编程）

原创

李四老师 于 2018-02-13 22:44:09 发布 1692 收藏 7

分类专栏：[#【8086汇编】](#) 文章标签：[汇编语言王爽](#) [实验9根据材料编程](#) [汇编语言hello world](#)

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_29542611/article/details/79323586

版权

8086 ASM

[【8086汇编】](#) 专栏收录该内容

15 篇文章 6 订阅
订阅专栏

实验介绍

学习每一种编程语言，第一个程序一般都是 **hello world**，一般都是往 **控制台进行输出**。但是学习汇编语言，汇编语言王爽这本书学习了一大半才有了往控制台输出内容。这是教材中的 **实验9 根据材料编程** 的实验，在 **屏幕中间分别显示黑底绿色、绿底红色、白底蓝色的字符串 'welcome to masm!'**。权且当做 **汇编语言的 hello world** 的吧。

实验结果

先展示运行程序运行结果

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
C:\>EXP9_1.EXE
C:\>
welcome to masm!
welcome to masm!
welcome to masm!
http://blog.csdn.net/qq_29542611
```

实验思路

其实就是往显存里面放数据！一个源数据段，一个目的数据段，一个临时数据段。显示3种不同颜色的字符串，可以用一个loop循环，

往显存中放字符串，同样需要一个loop，所以就要使用loop嵌套循环，使用嵌套循环那么cx就要存放到临时数据中。然后就是显示器每行80个字符，每个字符占2个字节，偶数字节存放字符ASCII码值，奇数字节存放显示属性。一定要明确数据在哪里，数据怎么处理，数据放哪里，这3个问题。

实验代码

;实验9 编程：在屏幕中间分别显示黑底绿色、绿底红色、白底蓝色的字符串'welcome to masm!'

```
assume cs:codesg
```

```
data segment
```

```
db 'welcome to masm!' ;16个字符
```

```
db 00000010B ;黑底绿字
```

```
db 00100100B ;绿底红字
```

```
db 01110010B ;白底绿字
```

```
data ends
```

```
stack segment
```

```
dw 0
```

```
stack ends
```

```
codesg segment
```

```
start:
```

```
;设置栈段 存放临时数据
```

```
mov ax,stack
```

```
mov ss,ax
```

```
mov sp,2
```

```
;往目的地址->显示器缓存区写东西就可以了
```

```
mov ax,0B800H
```

```
mov es,ax
```

```
;源地址
```

```
mov ax,data
```

```
mov ds,ax
```

```
mov si,16
```

```
mov di,160*11+160/2-16 ;每页25行，在11~13行进行居中展示。注意一个字符2个字节！
```

```
mov cx,3
```

```
showWelcom:
```

```
push cx
```

```
mov cx,16
```

```
mov bx,0
```

```
;每行80个字符，80个字节，显示属性占80个字节，一行占 160个字节
```

```
showString:
```

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

```
mov es:[di],al ;偶数字节 放字符ASCII值
```

```
mov al,ds:[si]
```

```
mov es:[di+1],al ;奇数字节 放颜色属性值
```

```
inc bx
```

```
add di,2
```

```
loop showString
```

```
add di,128
```

```
add si,1
```

```
pop cx
```

```
loop showWelcom
```

```
mov ax,4c00h
```

```
int 21h
```

```
codesg ends
```

```
end start
```

