电子科技大学信息与软件工程学院

实 验 报 告

学　　号 2018091618008

姓　　名 袁昊男

（实验）课程名称 汇编语言程序设计

理论教师 赵　洋

实验教师 赵　洋

电子科技大学教务处制表

电 子 科 技 大 学

实　验　报　告

学生姓名：袁昊男　　学号：2018091618008　　指导教师：赵洋

实验地点：在线实验　　实验时间：2020.05.11

一、实验室名称：信息与软件工程学院实验中心

二、实验名称：控制字符串在屏幕上的输出

三、实验学时：2 学时

四、实验原理：

8086CPU 支持 80×25 彩色字符模式，其显示缓冲区（以下简称为显示缓冲区）的结构如下：

内存地址空间中，B8000H～BFFFFH 共 32KB 的空间，为 80×25 彩色字符模式的显示缓冲区。向这个地址空间写入数据，写入的内容将立即出现在显示器上。

在 80×25 彩色字符模式下，显示器可以显示 25 行，每行 80 个字符，每个字符可以有 256 种属性（背景色、前景色、闪烁、高亮等组合信息）。这样，一个字符在显示缓冲区中就要占两个字节，分别存放字符的 ASCII 码和属性。80×25 模式下，一屏的内容在显示缓冲区中共占 4000 个字节。显示缓冲区分为 8 页，每页 4KB（≈4000B），显示器可以显示任意一页的内容。一般情况下，显示第 0 页的内容。也就是说通常情况下，B8000H～B8F9FH 中的 4000 个字节的内容将出现在显示器上。

在一页显示缓冲区中：

偏移 000～09F 对应显示器上的第 1 行（80 个字符占 160 个字节）；

偏移 0A0～13F 对应显示器上的第 2 行；

偏移 140～1DF 对应显示器上的第 3 行；

依此类推，可知，偏移 F00～F9F 对应显示器上的第 25 行。

在一行中，一个字符占两个字节的存储空间（一个字），低位字节存储字符的 ASCII 码，高位字节存储字符的属性。一行共有 80 个字符，占 160 个字节。

即在一行中：

00-01 单元对应显示器上的第 1 列；

02-03 单元对应显示器上的第 2 列；

04-05 单元对应显示器上的第 3 列；

依此类推，可知，9E～9F 单元对应显示器上的第 80 列。

例：在显示器的 0 行 0 列显示黑低绿色的字符串' ABCDEF '（'A'的 ASCII 码值为 41H，02H 表示黑底绿色）

显示缓冲区里的内容为：

```
           00  01  02  03  04  05  06  07  08  09  0A  0B  ... 0E  0F
B800:0000  41  02  42  02  43  02  44  02  45  02  46  02  ... ..  ..
  :
  :
B800:00A0  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ... ..  ..
```

可以看出，在显示缓冲区中，偶地址存放字符，奇地址存放字符的颜色属性。一个在屏幕上显示的字符，具有前景（字符色）和背景（底色）两种颜色，字符还可以以高亮度和闪烁的方式显示。前景色、背景色、闪烁、高亮等信息被记录在属性字节中。属性字符的含义如下：

	7	6	5	4	3	2	1	0
含义	BL	R	G	B	I	R	G	B
	闪烁	背景			高亮	前景		

R：红色

G：绿色

B：蓝色

可以按位设置属性字节，从而配出各种不同的前景色和背景色。

比如：

红底绿字，属性字节为：01000010B；

红底闪烁绿字，属性字节为：11000010B；

红底高亮绿字属性字节为：01001010B；

黑底白字，属性字节为：00000111B；

白底蓝字，属性字节为：0111l0001B。

例：在显示器的 0 行 0 列显示红底高亮闪烁绿色的字符串'ABCDEF'（红底高亮闪烁绿色，属性字节为：11001010B，CAH）

显示缓冲区里的内容为：

```
           00  01  02  03  04  05  06  07  08  09  0A  0B  ... 9E  9F
B800:0000  41  CA  42  CA  43  CA  44  CA  45  CA  46  CA  ... ..  ..
  :
  :
B800:00A0  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ... ..  ..
```

注意，闪烁的效果必须在全屏 DOS 方式下才能看到。

五、实验目的：

1、 掌握 8086 显示缓存的工作模式。

2、 掌握通过 8086 显示缓存控制字符串在屏幕上的输出。

六、实验内容：

编程实现：在屏幕中间分别显示绿色、绿底红色、白底蓝色的字符串' welcome to masm! '。

七、实验器材（设备、元器件）：

PC 微机一台

八、实验步骤：

1、 编辑源程序，建立一个以后缀.ASM 的文件。

2、 汇编源程序，检查程序有否错误，有错时回到编辑状态，修改程序中错误行，无错时继续第 3 步。

3、 连接目标程序，产生可执行程序。

4、 用 DEBUG 程序调试可执行程序，记录数据段的内容。

九、实验数据及结果分析

1、 实验思路

(1) 总体思路

将字符串' welcome to masm! '与所要求的三种字符属性字节依次连续存储在 data 段中；使用嵌套的循环结构，并辅以栈的操作来实现将字符 ASCII 码字节与对应字符属性字节组装后送入 80×25 彩色字符模式的显示缓冲区中对应单元，使得程序在屏幕中间输出三行不同颜色的字符串。

(2) 字符属性设置

字符属性占一字节，由 8 位二进制数构成。因此，绿色字符属性为 00000010（02H），绿底红色字符属性为 00100100（24H），白底蓝色字符属性为 01110001（71H）。

(3) 显示缓冲区中存储地址

由于要求在屏幕中间输出三行不同颜色的字符串，因此必须保证这三行字符串水平居中、垂直居中。在 80×25 彩色字符模式下，显示器可以显示 25 行，每行 80 个字符，字符串' welcome to masm! '共 16 个字符，由 $(80-16)\div 2=32$、$(25-3)\div 2=11$可得三行字符串左右各空余 32 列字符、上下各空余 11 行字符。因此第一行第一个字符所在位置应该是第 12 行、33 列（均从 1 开始计数）。一个字符占 2 字节，一行共有 80 个字符，占

160 个字节；11 行共占11×160＝1760（6E0H）个字节，因此第 12 行起始地址为 B800:06E0（0 开始计数）；行内起始单元地址为第 64 号单元字节（左边空余 32 个字符，共 64 字节，0 开始计数）。

(4) 循环嵌套

采用双重循环：外循环共执行 3 次，控制行的改变；内循环共执行 16 次，控制列的改变。每次内循环首先分别将地址指针定位于显示缓冲区第一个字符输出位与字符串第一个字符处，然后将字符 ASCII 码字节存入显示缓冲区对应存储器的低字节单元，将字符属性字节存入显示缓冲区对应存储器的高字节单元。完成一个字符操作后，显示缓冲区指针向后移动 2 字节，data 段字符串指针向后移动 1 字节。双重循环还需要栈操作辅助，外层循环执行完成后将 cx、si 寄存器值先后压栈保护；内层循环执行完成后再将 si、cx 寄存器值先后弹栈恢复（注意先后顺序）。

2、 流程图

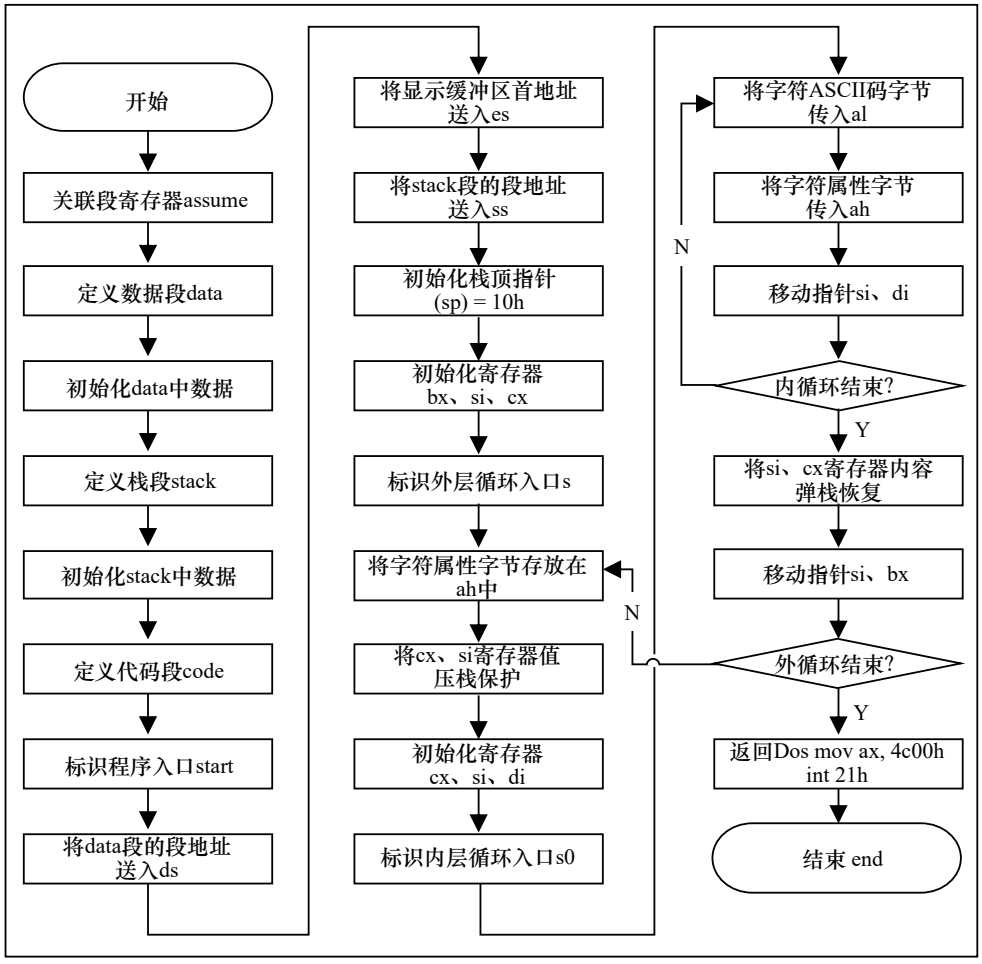

图 1　实验二流程图

3、 实验截图

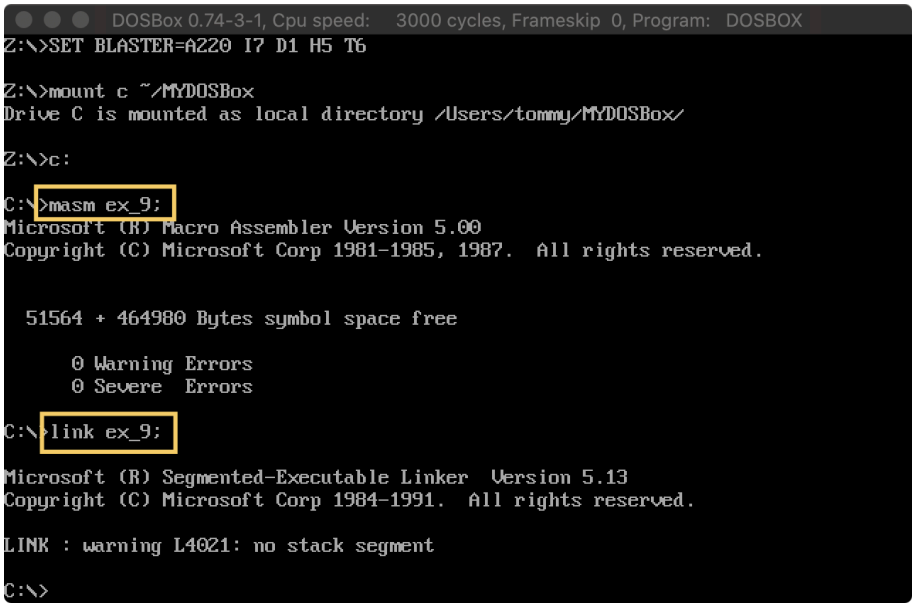

图 2　编译、连接

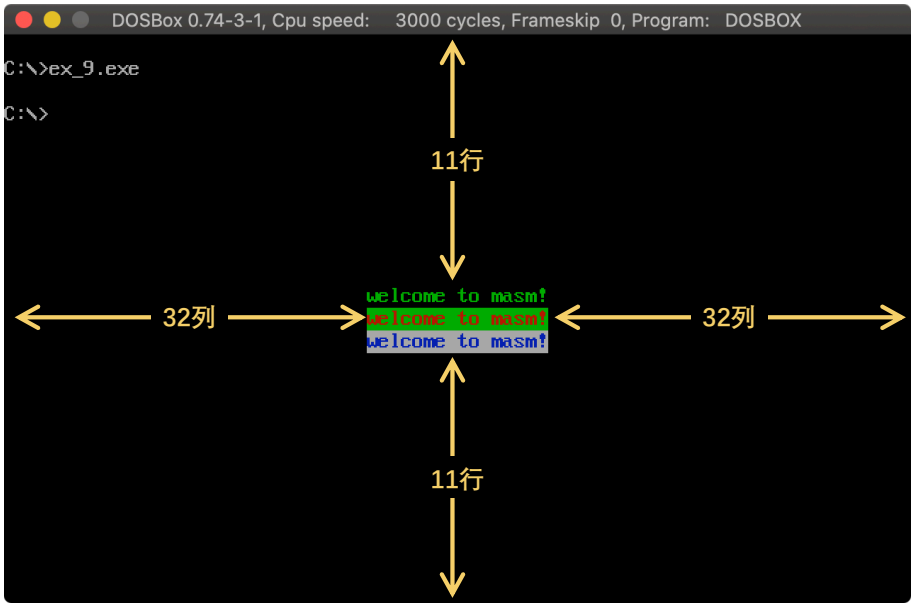

图 3　运行结果（正确在屏幕中间显示三行字符串）

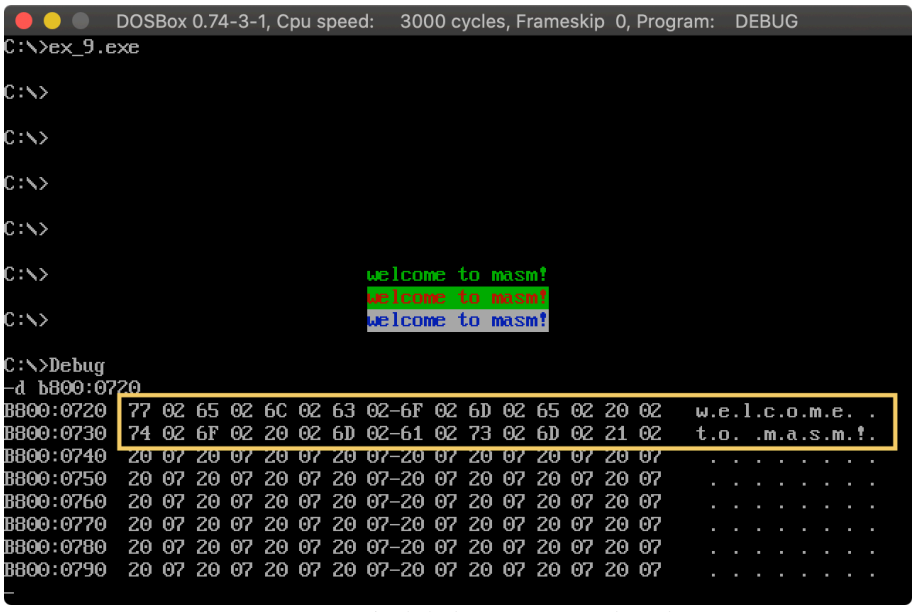

图 4　第一行字符串在显示缓冲区中的存储

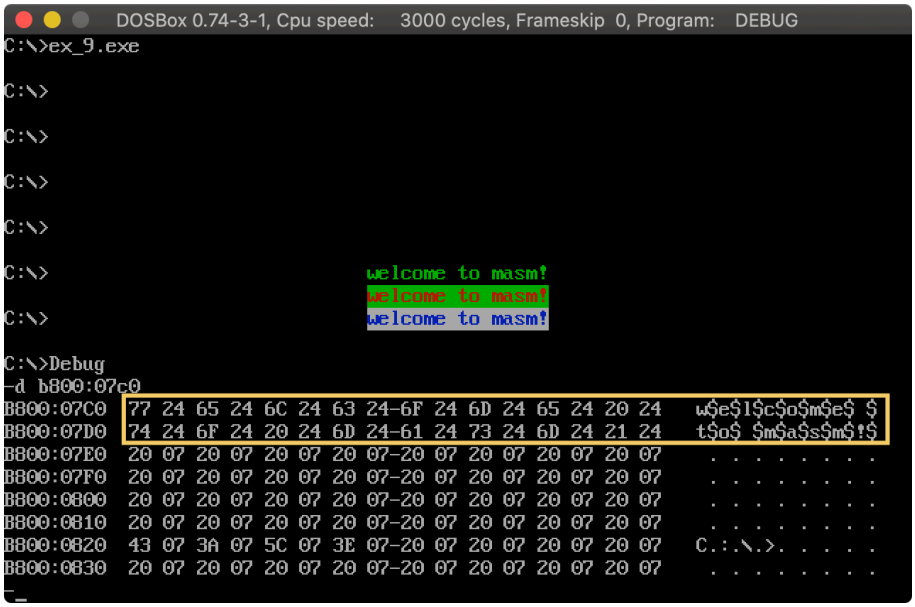

图 5　第二行字符串在显示缓冲区中的存储

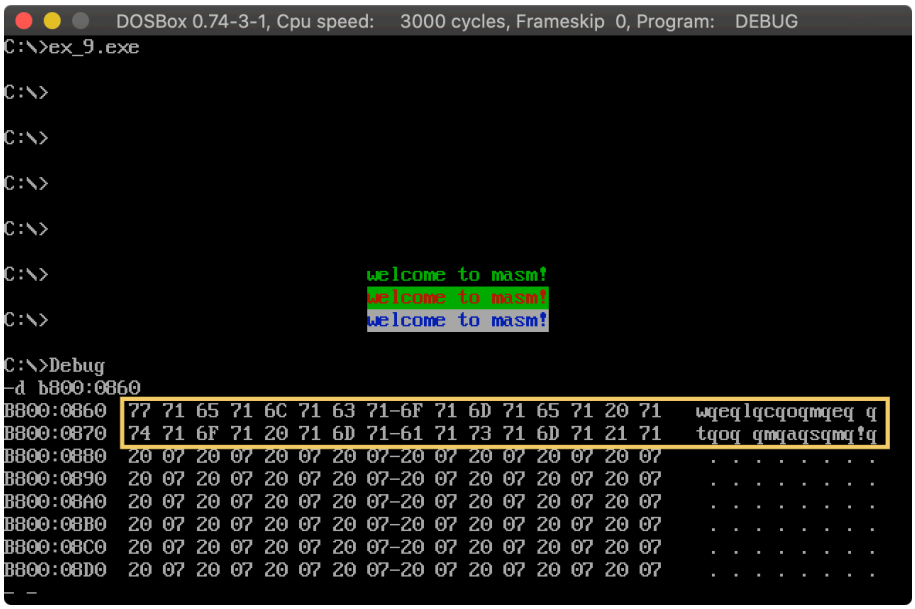

图 6　第三行字符串在显示缓冲区中的存储

4、 结果分析

程序能正确编译、连接，运行后可以按实验要求在屏幕中间分别显示绿色、绿底红色、白底蓝色的字符串。经跟踪、调试，使用 D 命令查看显示缓冲区存储数据（如图 4、图 5、图 6 所示），可以发现三行字符串对应编码在显示缓冲区中按预设内容存储：绿色行是字符 ASCII 码+02H、绿底红色行是字符 ASCII 码+24H、白底蓝色行是字符 ASCII 码+71H。

十、实验结论

代码经 masm 编译、link 连接成功，运行后可以按实验要求在屏幕对应位置正确显示不同字符属性的字符串。经 Debug 跟踪、调试，在显示缓冲区中字符数据

及字符属性数据按预设正确存储。

由此可得出：汇编语言可以通过控制字符属性编码及地址偏移量可以对显示缓冲区数据进行操作，从而实现在屏幕指定位置输出指定格式的字符。

十一、总结及心得体会

1、 在 80×25 彩色字符模式的显示缓冲区中，一个字符占 2 字节（ASCII 码与字符属性编码各占 1 字节），每个字符的输出可实现背景色、前景色、闪烁、高亮等不同效果。显示缓冲区地址范围为 B8000H～B8F9FH，共 4000 字节，可以根据字符需要显示的位置来计算其在显示缓冲区中的对应地址，并将字符 ASCII 编码与字符属性编码组合后存入对应地址，即可实现输出要求。

2、 在使用 Debug 进行跟踪、调试时，由于显示缓冲区中的内容存储的是当前屏幕所显示的实时内容，因此如果要查看三行字符串在显示缓冲区中的存储情况，采用换行或清屏功能将屏幕干扰信息清空，并计算字符串的起始地址，直接使用 D 命令查看该地址内容。或者可在程序结束前将显示缓冲区中的内容复制到另一数据段中，再使用 D 命令对该数据段进行查看。

十二、对本实验过程及方法、手段的改进建议

本实验的讲解十分细致，还可以在实验内容上还可以增加拓展性内容，如“自行设计多变化的字符串输出”等，可以提高学生参与实验的积极性，加深对本实验内容的理解与掌握。

报告评分：

指导教师签字：

附录：实验程序源码

```
assume ds:data,cs:code,es:stack

data segment
    db 'welcome to masm!'
    db 02h,24h,71h              ; 要求的三个颜色对应的 16 进制代码
data ends

stack segment
    db 16 dup(0)
stack ends

code segment
start:
       mov ax,data
       mov ds,ax                ; 将 data 段地址送入 ds
       mov ax,0b800h
       mov es,ax                ; 将显示缓冲区起始地址送入 es

       mov ax,stack
       mov ss,ax                ; 将 stack 段地址送入 ds
       mov sp,10h               ; 指向栈顶

       mov bx,6e0h              ; 定位至第一行字符串输出位置
       mov si,10h               ; 定位至 data 段字符属性数据

       mov cx,3                 ; 三次外循环改变行
   s:  mov ah,ds:[si]           ; 字符属性先存放在 ah 中

       push cx
       push si                  ; 压栈保护

       mov cx,16                ; 16 次内循环改变列
       mov si,64                ; 定位至每一行字符串的第一列输出位置
       mov di,0
   s0: mov al,ds:[di]
       mov es:[bx+si],al        ; 将字符 ASCII 码传入 al
       mov es:[bx+si+1],ah      ; 高位存放字符属性

       add si,2                 ; 显示缓冲区字符 ASCII 码偏移量为 2
       add di,1                 ; data 段字符的偏移量为 1

       loop s0

       pop si
       pop cx                   ; 后进先出

       add si,1h                ; 指向下一个字符属性
       add bx,0a0h              ; 指向下一行
       loop s

       mov ax,4c00h
       int 21h
code ends

end start
```