# 

**计算机科学与技术学院**

**2016-2017学年第2学期**

**《汇编语言程序设计》**

**实验2：汇编语言上机过程**

**专业： 信息安全**

**学号： 150341221**

**姓名： 路伟饶**

**教师： 王家亮**

**完成日期：2017年4月1日**

1. 实验要求
2. 列出源程序清单
3. 分析实验中所遇到的一些问题，分析错误原因
4. 说明本实验中是如何使用 DEBUG 进行调试的
5. 写出本次上机的体会或收获
6. 实验目的
7. 掌握汇编语言源程序的结构
8. 熟悉汇编语言源程序的编辑、汇编、连接、调试过程
9. 理解和掌握变量的定义方法，系统如何为变量分配空间
10. 理解和掌握汇编指令(LEA、ADD)的功能和用法
11. 程序说明
12. 已知有以下变量定义，请将各变量分别放在源程序中进行调试，观察为每变量所分配的存储空间及初始化的数据值。

DATA SEGMENT

BR DB 'Hello',68,-20,3 DUP(4) ;按Byte分配变量BR

WR DW 3456H,0AFH,0A123H ;按Word分配变量WR

X DW 1,2,$+4,3,4,$+4 ;按Word分配变量X

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA ;将DATA的值装入AX

MOV DS,AX ;将AX装入DS寄存器

LEA DX,BR ;将变量BR的偏移地址装入DX

MOV AH,9 ;设置9号中断

INT 21H ;启动中断

LEA DX,WR ;将变量WR的偏移地址装入DX

MOV AH,9 ;设置9号中断

INT 21H ;启动中断

LEA DX,X ;将变量X的偏移地址装入DX

MOV AH,9 ;设置9号中断

INT 21H ;启动中断

MOV AX,4C00H

INT 21H

CODE ENDS

END START

1. 执行下列指令后，AX寄存器中的内容是什么？将以上程序段补充成完整的汇编源程序，并调试运行。

DATA SEGMENT

TABLE DW 10H,20H,30H,40H,50H ;按Word分配变量TABLE

BUF DW 4 ;按Word分配变量BUF

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA

MOV DS,AX

LEA BX,TABLE ;将TABLE的偏移地址装入BX

ADD BX,BUF ;相当于BX=TABLE的偏移地址+4

MOV AX,[BX] ;将BX偏移地址对应内存的内容装入AX

;AX应为0030H

LEA DX,[BX] ;将BX中保存的偏移地址装入DX

MOV AH,9 ;设置9号中断

INT 21H ;启动中断

MOV AX,4C00H

INT 21H

CODE ENDS

END START

1. 将以上程序段中的TABLE DW 10H,20H,30H,40H,50H修改为TABLE DW 10,20,30,40,50再补充成完整的汇编程序并调试运行，观察并记录 AX 的内容。

DATA SEGMENT

TABLE DW 10,20,30,40,50 ;按Word分配变量TABLE

BUF DW 4 ;按Word分配变量BUF

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA

MOV DS,AX

LEA BX,TABLE ;将TABLE的偏移地址装入BX

ADD BX,BUF ;相当于BX=TABLE的偏移地址+4

MOV AX,[BX] ;将BX偏移地址对应内存的内容装入AX

;AX应为0030D，也就是001EH

LEA DX,[BX] ;将BX中保存的偏移地址装入DX

MOV AH,9 ;设置9号中断

INT 21H ;启动中断

MOV AX,4C00H

INT 21H

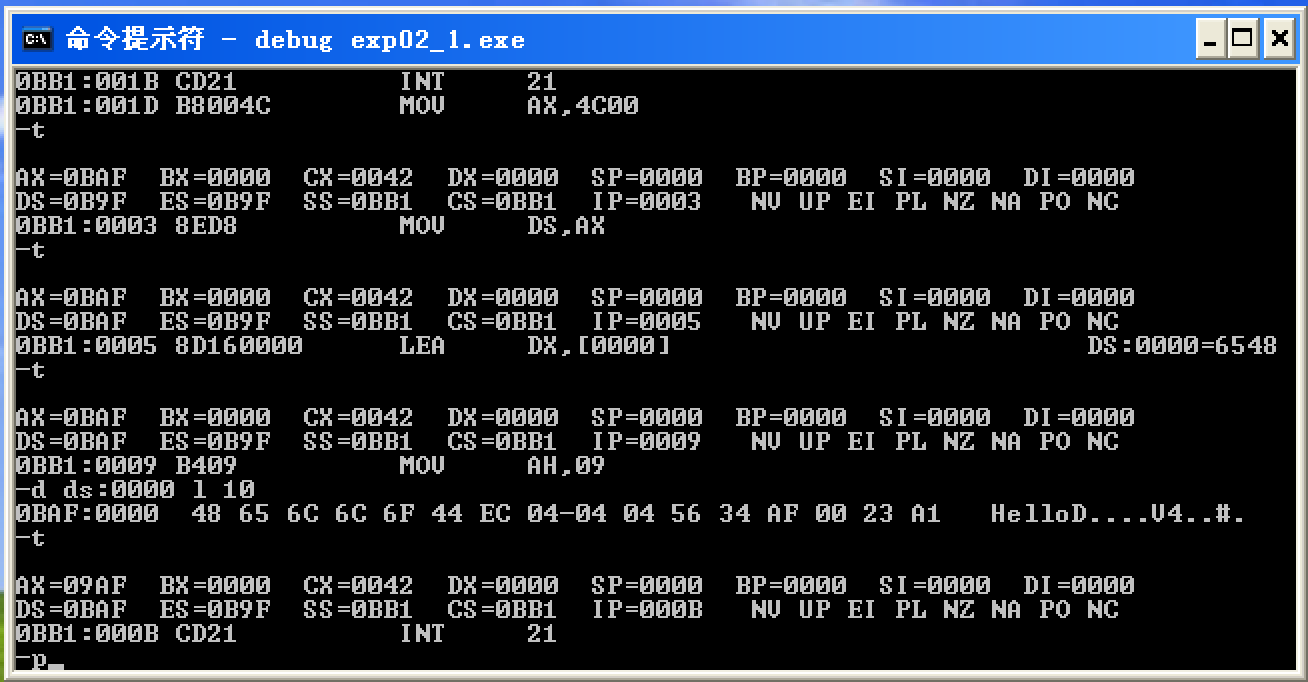
CODE ENDS

END START

1. 上机调试说明

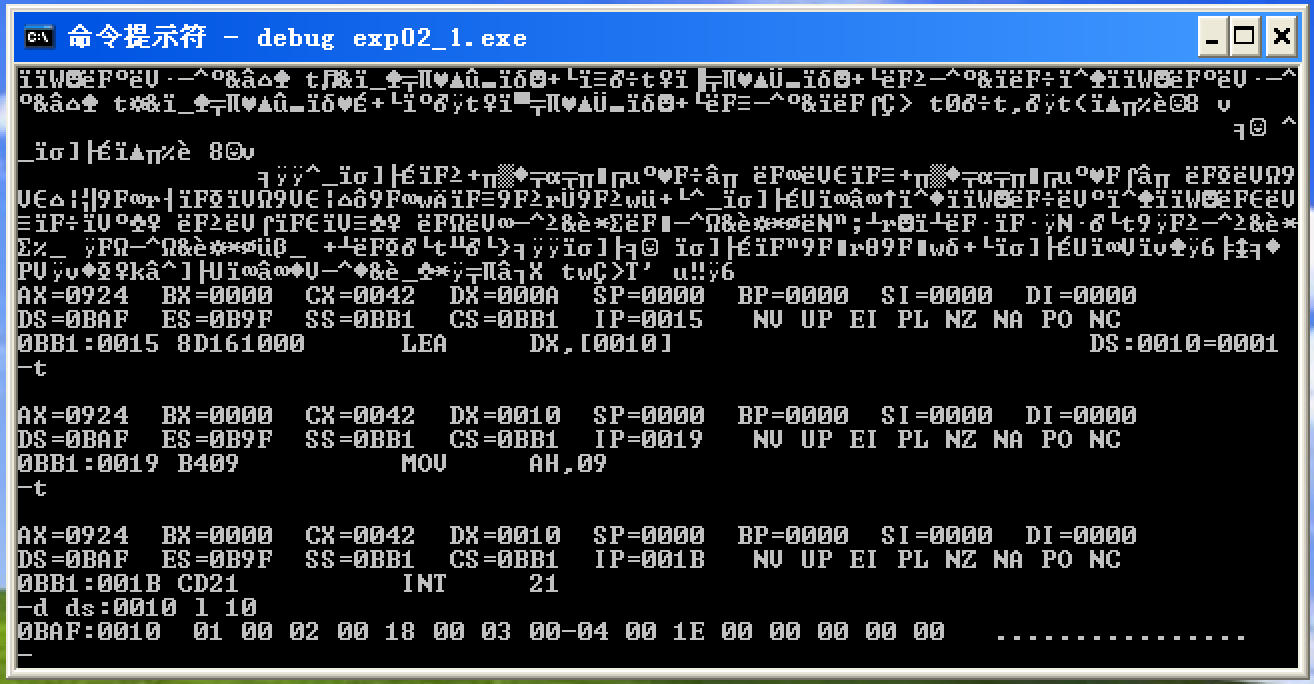
本次实验的源程序使用Windows XP虚拟机中的MASM进行编译连接和运行，无需Dosbox，在操作流程上比较容易。在调试过程中关于实验内容没有遇到明显的问题，不过有一个问题就是使用9号中断输出字符串的值的时候会出来很长的一串，远远超出了我们变量存放的地址范围，类比C语言的字符串处理，推测是没有遇到‘\0’之类的问题造成的，稍后会进行探究。

1. 写出源程序清单及执行结果
2. 已知有以下变量定义，请将各变量分别放在源程序中进行调试，观察为每变量所分配的存储空间及初始化的数据值。



按程序定义，通过d命令查看ds:0000的内存，可以看出内存中存放了变量BR、WR的值（X的值在下一张图）。

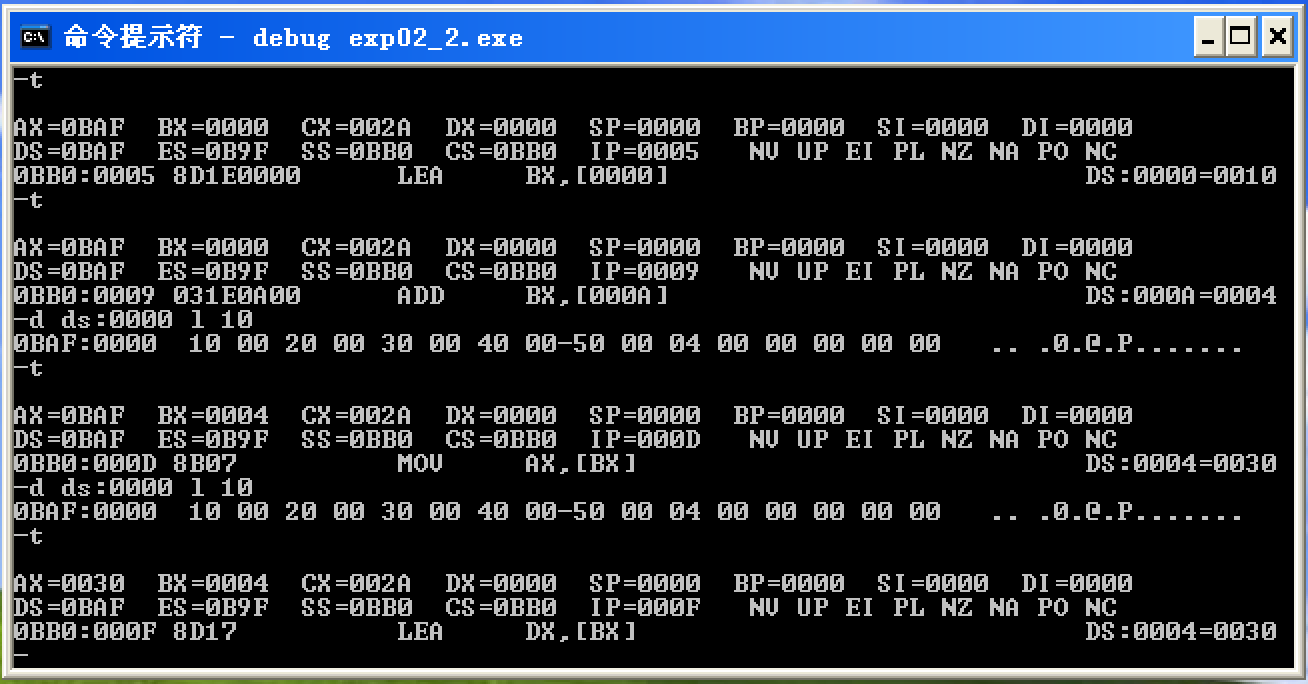
|  |  |  |
| --- | --- | --- |
| 内存地址 | 存放内容 | 原始值 |
| 0000H | 48H | ‘H’ |
| 0001H | 65H | ‘e’ |
| 0002H | 6CH | ‘l’ |
| 0003H | 6CH | ‘l’ |
| 0004H | 6FH | ‘o’ |
| 0005H | 44H | 68 |
| 0006H | 0ECH | -20 |
| 0007H | 04H | 4 |
| 0008H | 04H | 4 |
| 0009H | 04H | 4 |
| 000AH | 56H | 3456H |
| 000BH | 34H |
| 000CH | 0AFH | 0AFH |
| 000DH | 00H |
| 000EH | 23H | 0A123H |
| 000FH | A1H |



内存ds:0010区域存放了变量X的值

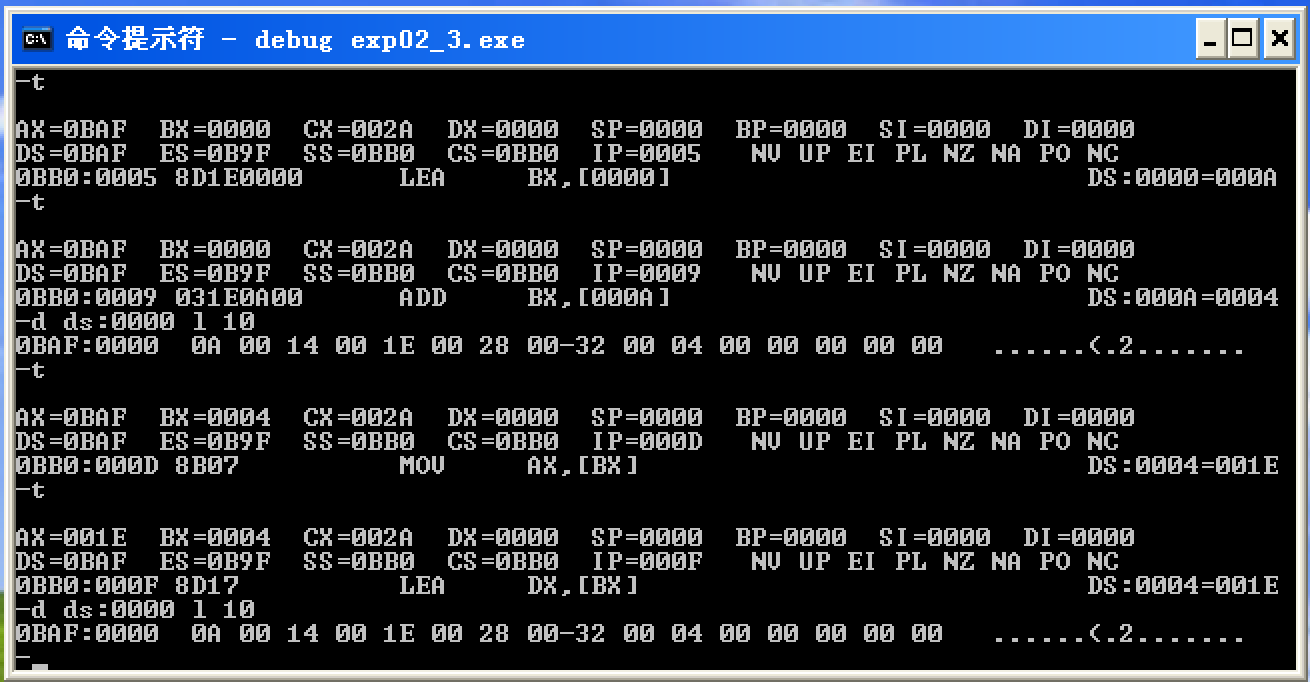
|  |  |  |
| --- | --- | --- |
| 内存地址 | 存放内容 | 原始值 |
| 0010H | 01H | 1 |
| 0011H | 00H |
| 0012H | 02H | 2 |
| 0013H | 00H |
| 0014H | 18H | $+4  其中$=0014H |
| 0015H | 00H |
| 0016H | 03H | 3 |
| 0017H | 00H |
| 0018H | 04H | 4 |
| 0019H | 00H |
| 001AH | 1EH | $+4  其中$=001AH |
| 001BH | 00H |
| 001CH | 00H | 未使用内存 |
| 001DH | 00H |
| 001EH | 00H |
| 001FH | 00H |

1. 执行下列指令后，AX寄存器中的内容是什么？将以上程序段补充成完整的汇编源程序，并调试运行。



程序取的是TABLE的偏移地址+4的内存中的内容，也就是我们通过变量TABLE定义的03H。

1. 将以上程序段中的TABLE DW 10H,20H,30H,40H,50H修改为TABLE DW 10,20,30,40,50再补充成完整的汇编程序并调试运行，观察并记录 AX 的内容。



程序取的是TABLE的偏移地址+4的内存中的内容，也就是我们通过变量TABLE定义的03，十六进制为1E。

1. 心得体会

第二次上机的内容也不复杂，主要是熟悉MOV、LEA、ADD指令的使用，以及关于伪代码的内容，主要是变量的定义，方式可以是DB也就是按字节定义和存放，也可以是DW方式，也就是按字定义和存放，因为8086CPU是16位的，所以一个Word正好是2个字节，同时也了解了系统为变量分配内存空间的方式。在debug程序的过程中，再次熟悉了-u -t -p -d这几个debug指令的使用方法和应用范围。

1. 关于9号中断的一些探究

为了直观查看内存中的值，在本次实验中我利用了一下9号中断，这样就可以查看对应内存中的值，但是在运行的时候发现了一些问题。程序输出了很长的一段内容，类似于C语言的字符数组越界，源程序和运行效果如下：

DATA SEGMENT

STR DB 'Hello',68,-20,3 DUP(4);按Byte分配变量BR

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA ;将DATA的值装入AX

MOV DS,AX ;将AX装入DS寄存器

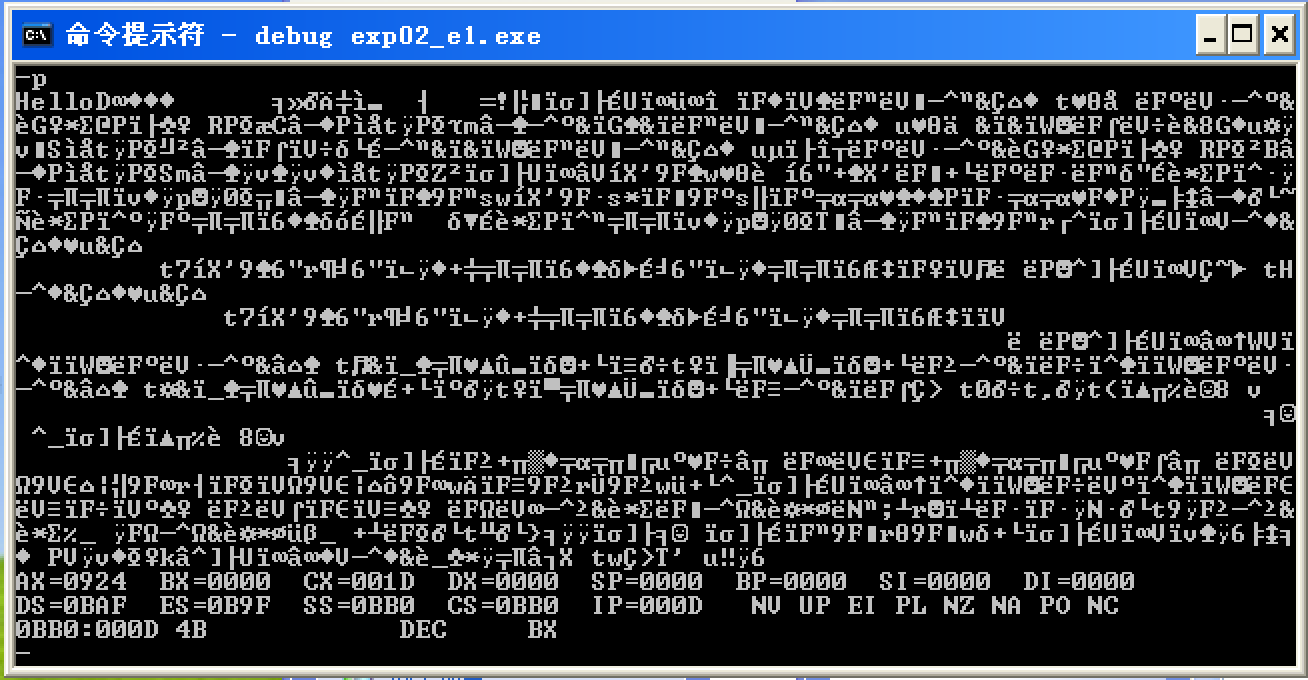
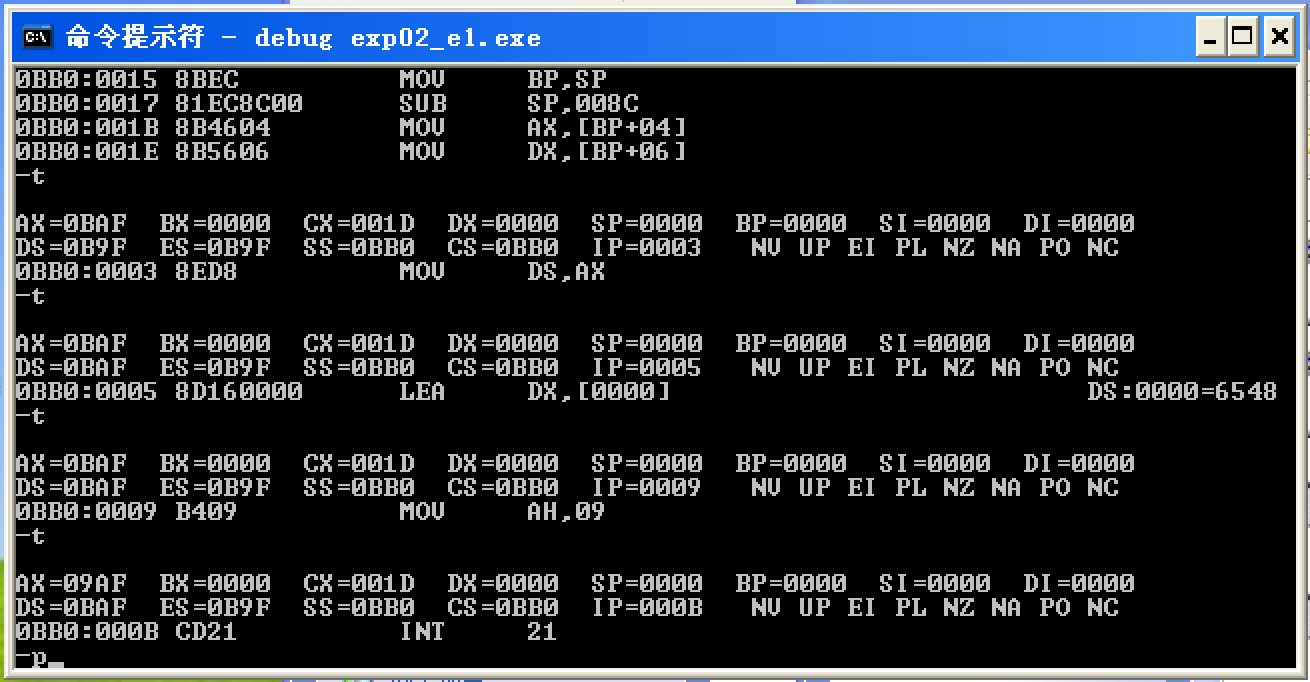
LEA DX,STR ;将变量STR的偏移地址装入DX

MOV AH,9 ;设置9号中断

INT 21H ;启动中断

CODE ENDS

END START

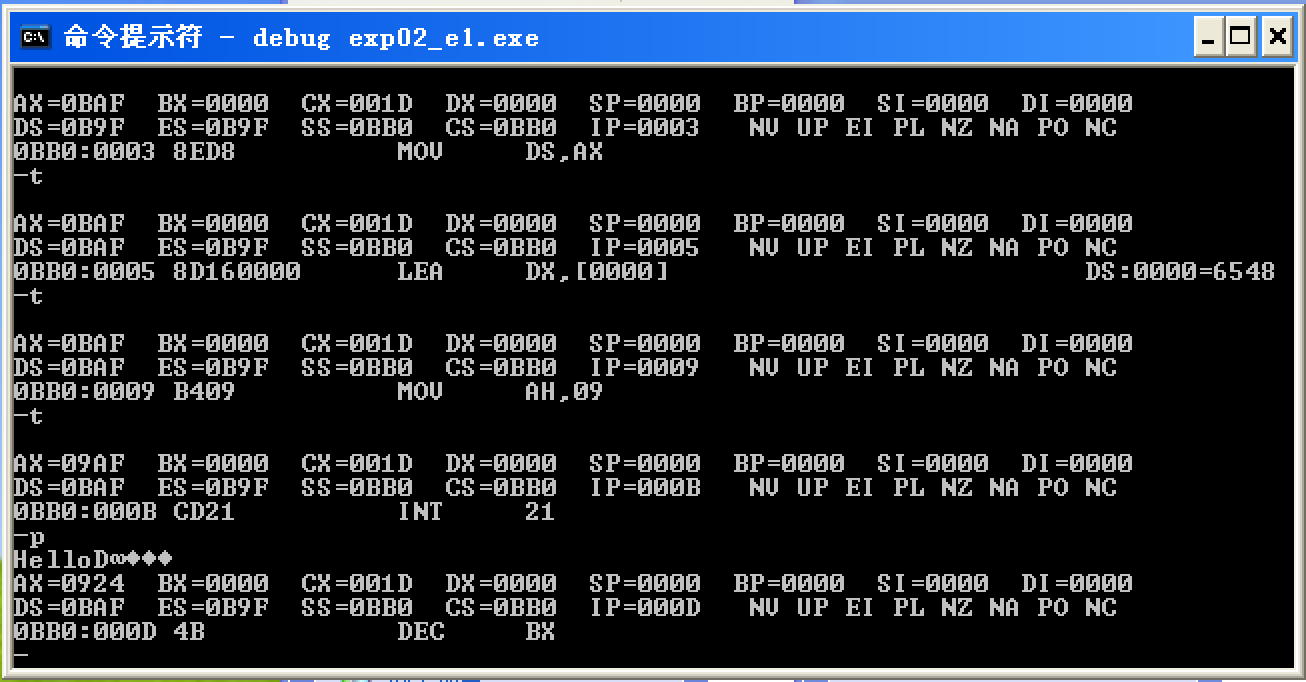


推测是因为没有加\0之类的东西导致的，后来发现课上演示程序的末尾有一个$，加上就真的解决了问题：

DATA SEGMENT

STR DB 'Hello',68,-20,3 DUP(4),'$';按Byte分配变量BR

DATA ENDS



这样就只输出到$处便停止了，但是在平日里另一个写的程序里，使用了课上程序的0AH，功能为换行，但是这个0AH的换行功能和我们预想的略有不同，源程序和运行结果如下：

DATA SEGMENT

STR DB 0AH,'Hello World!$'

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

CODE SEGMENT

ASSUME DS:DATA,SS:STACK,CS:CODE

START:

MOV AX,DATA

MOV DS,AX

LEA DX,STR

MOV CX,04H

s: MOV AH,9

INT 21H

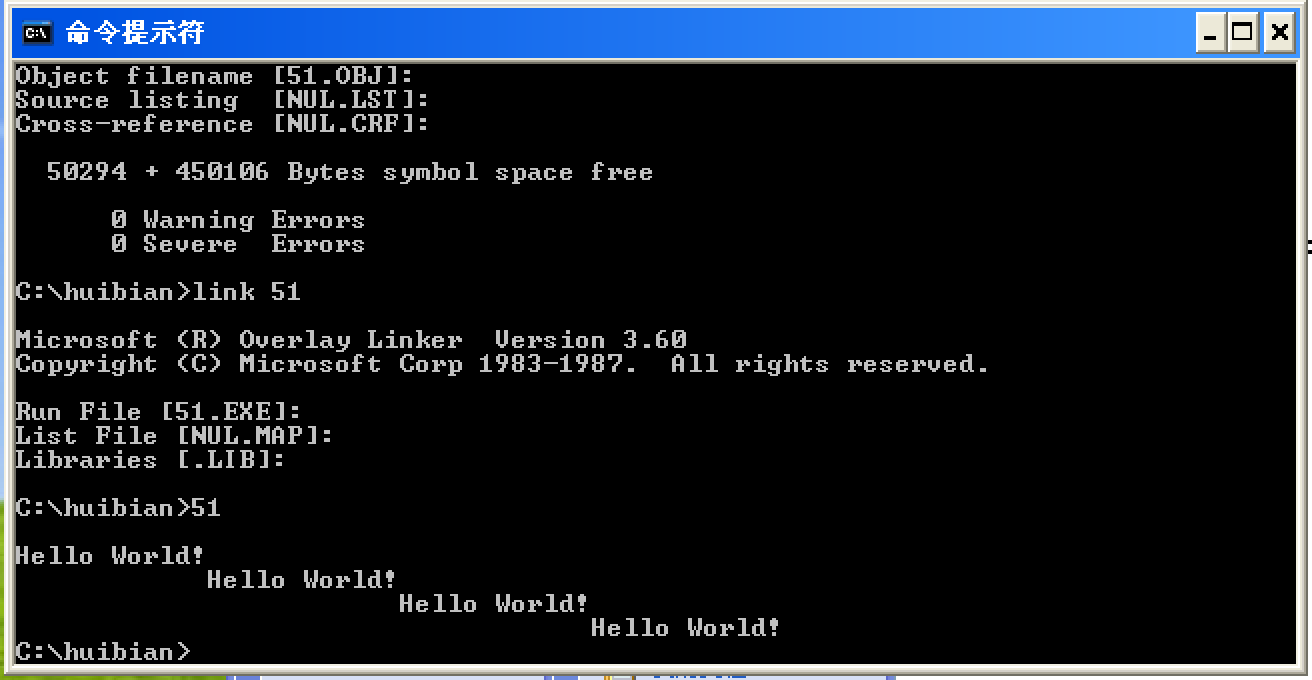
loop s

MOV AX,4C00H

INT 21H

CODE ENDS

END START



因为CX=4，所以使用9号中断程序输出四次Hello World到屏幕。（将MOV CX,04H改为MOV CX,0FFFFH有更奇幻的效果）但是这个好像是换行的时候还和上面的末尾进行对齐了，具体原因还有待探究。



MOV CX,0FFFFH的情况，迷之奇幻效果。