

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验一 编程基础**

**实验时间： 2017-10-18，18：30-21：50 实验地点： 南一楼804室**

**指导教师： 朱虹**

**专业班级：信息安全 201602班**

**学 号： U201614836 姓 名： 林晓斌**

**同组学生： 无 报告日期： 2017年 10 月 18日**

**原创性声明**

本人郑重声明：本报告的内容由本人独立完成，有关观点、方法、数据和文献等的引用已经在文中指出。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品或成果，不存在剽窃、抄袭行为。

特此声明！

学生签名：

日期：2017.10.17

|  |  |  |
| --- | --- | --- |
| 实验完成质量得分（70分）（实验步骤清晰详细深入，实验记录真实完整等） | 报告撰写质量得分（30分）（报告规范、完整、通顺、详实等） | 总成绩（100分） |
|  |  |  |

指导教师签字：

                     日期：

目录

[实验一 3](#_Toc500881542)

[1.1 实验目的与要求 3](#_Toc500881543)

[1.2 实验步骤 3](#_Toc500881544)

[1.3 实验内容 3](#_Toc500881545)

[1.3.1 任务1 4](#_Toc500881546)

[1.3.2 任务2 6](#_Toc500881547)

[1.3.3 任务3 8](#_Toc500881548)

[1.3.4 任务4 11](#_Toc500881549)

[1.4 实验小结 13](#_Toc500881550)

[1.5参考文献 13](#_Toc500881551)

# 实验一

# 1.1 实验目的与要求

1. 掌握汇编源程序编辑工具、汇编程序、连接程序、调试工具TD的使用；
2. 理解数、符号、寻址方式等在计算机内的表现形式；
3. 理解指令执行与标志位改变之间的关系；
4. 熟悉常用的DOS功能调用；
5. 熟悉分支、循环程序的结构及控制方法，掌握分支、循环程序的调试方法；
6. 加深对转移指令及一些常用的汇编指令的理解。

# 1.2 实验步骤

1. 通过寄存器间接寻址，变址寻址，基址加变址寻址，直接寻址，分配一数据段和段内变量，以立即数方式写入内存的相应位置。

2. 使用编辑程序EDIT.EXE录入源程序，存盘文件名为DEMO.ASM。使用MASM 6.0汇编源文件。即MASM DEMO；观察提示信息，若出错，则用编辑程序修改错误，存盘后重新汇编，直至不再报错为止。

3. 使用连接程序LINK.EXE将汇编生成的CUBE.OBJ文件连接成执行文件。若连接时报错，则依照错误信息修改源程序。之后重新汇编和连接，直至不再报错并生成.DEMO.EXE文件。

4. 使用TD.EXE观察程序的执行情况。即 TD DEMO.EXE回车

（1）运行到“INT 21H”。

（2）在数据窗口中观察各项数据的值。

# 1.3 实验内容

任务1. 内存单元的访问。

以四种不同的内存寻址方式，将自己学号的后四位依次存储到 以 XUEHAO开头的存储区中。

要求：在报告中给出完整的程序；给出运行效果截图；（不需要画流程图）；在程序注释中，明确指出访问存储单元时，用的是什么寻址方式。

任务2. 《80X86汇编语言程序设计》教材中 P45的 2.3题。

要求：（1）分别记录执行到“MOV CX，10”和“INT 21H”之前的(BX), (BP),(SI),(DI)各是多少。

（2）记录程序执行到退出之前数据段开始40个字节的内容，指出 程序运行结果是否与设想的一致。

任务3. 《80X86汇编语言程序设计》教材中 P45的 2.4题的改写。

要求：(1) 实现的功能不变，对数据段中变量访问时所用到的寻址方式中的寄存器改成32位寄存器。

(2) 内存单元中数据的访问采用变址寻址方式。

(3) 记录程序执行到退出之前数据段开始40个字节的内容，检查 程序运行结果是否与设想的一致。

(4)在TD代码窗口中观察并记录机器指令代码在内存中的存放形式，并与TD中提供的反汇编语句及自己编写的源程序语句进行对照，也与任务2做对比。（相似语句记录一条即可，重点理解机器码与汇编语句的对应关系，尤其注意操作数寻址方式的形式）。

(5)观察连续存放的二进制串在反汇编成汇编语言语句时，从不同字节位置开始反汇编，结果怎样？理解 IP/EIP指明指令起始位置的重要性。

任务4. 选做题

设有 DATA SEGMENT USE16

MSG1 DB ‘Hello 123’

LEN = $ - MSG1 ; MSG1中字符的个数

MSG2 DB LEN DUP ( 0 )

DATA ENDS

写一个程序，将MSG1中的串逆序后存储到变量MSG2中。

1.3.1 任务1

1.3.1.1 设计思想及存储单元分配

建立一个新的存储区域以‘XUEHAO’来命名，并为其分配四个字节。取出存储区域‘XUEHAO’的首地址，并将其传送到BX寄存器。分别将‘4836’四个字符传送到‘XUEHAO’存储单元中。

1 存储单元分配

XUEHAO:字节变量XUEHAO中存放‘4836’四个字符。

2 寄存器分配

BX: 用于存放XUEHAO变量的初始地址，也用于寄存器寻址以及变址寻址方式。

AL: 用于接受字符串并转送给存储单元

SI: 用于作为基址加变址寻址方式中的变址寄存器

1.3.1.2 源程序

;----------------------------

.386

STACK SEGMENT STACK USE16

DB 200 DUP(0)

STACK ENDS

;----------------------------

DATA SEGMENT USE16

XUEHAO DB 4 DUP(0)

DATA ENDS

;------------------------------

CODE SEGMENT USE16

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

BEGIN: MOV AX,DATA

MOV DS,AX

MOV BX,OFFSET XUEHAO

MOV AL,'4'

MOV [BX],AL ;寄存器间接寻址

MOV AL,'8'

MOV [BX]+1,AL ;变址寻址

MOV SI,2

MOV AL,'3'

MOV [BX][SI],AL ;基址加变址寻址

MOV AL,'6'

MOV DS:[3H],AL ;直接寻址

MOV AH,4CH ;EXIT

INT 21H

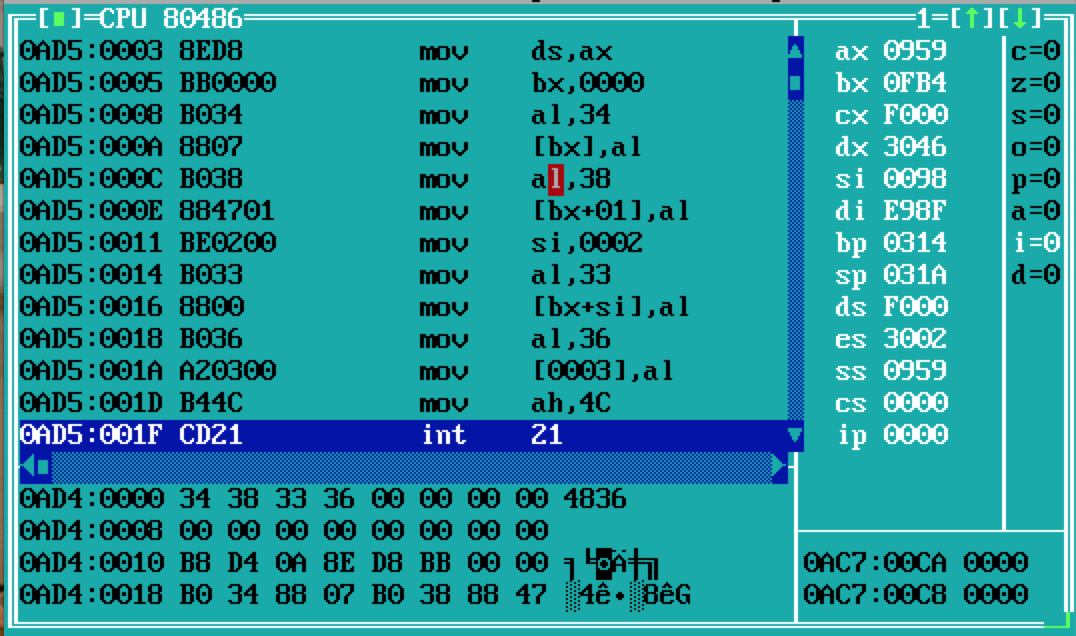
;-----------------------------

CODE ENDS

END BEGIN

1.3.1.3 运行效果截图

程序运行编译成功，在td中运行效果如图3.1.

图1.1程序运行效果图

1.3.2 任务2

1.3.2.1 实验步骤

1 准备上机实验环境

2 在“MOV CX，10”和“INT 21H”之前设置断点，当运行到该语句之时，程序会自动停止运行，此时观察并记录侧边栏的(BX),(BP),(SI),(DI)的值。

3 取消断点，运行程序，记录下数据段开始40个字节的内容，并与设想结果进行比较。

1.3.2.2 实验记录与分析

1 实验环境条件：2.9GHz Intel Core i5 , 8G内存，macOS Sirra下的DOSbox 0.74; TD.exe6.0

2.在“MOV CX，10”和“INT 21H”之前分别插入断点，当程序运行到“MOV CX，10”时，BX =0014 BP=001E DI=000A SI=0000 ，程序运行效果截图如图3.2。当程序运行到“INT 21H”时，BX =FEAE BP=0314 DI=3046 SI=F000 ，程序运行效果截图如图3.3.当程序运行完毕时，数据段开始前40字节内容如图3.4，图3.5所示。

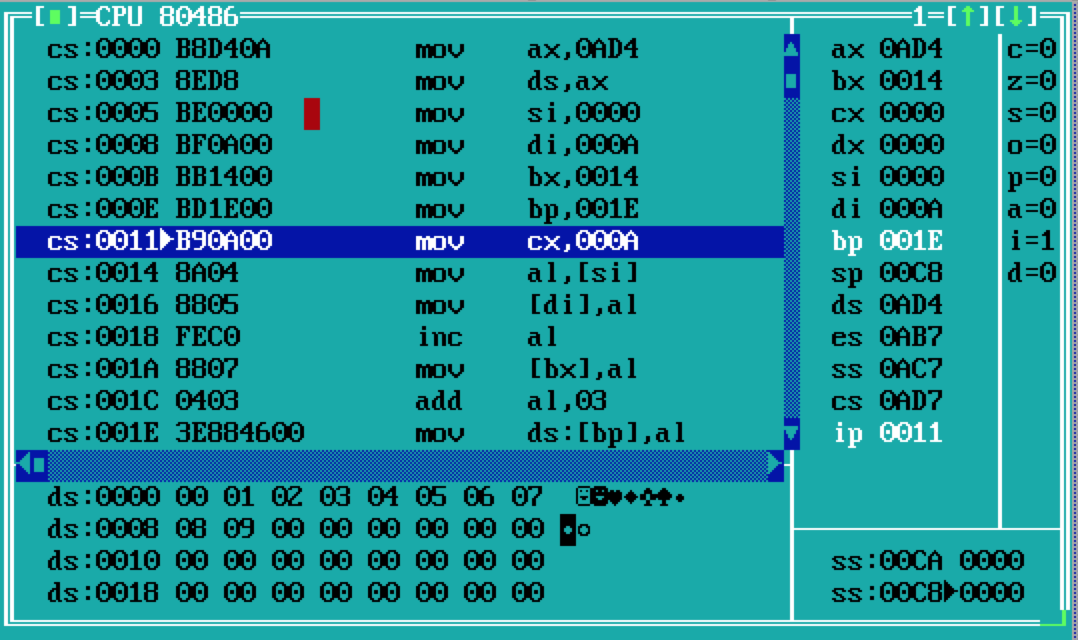


图1.2运行到“MOV CX，10”时程序运行效果图

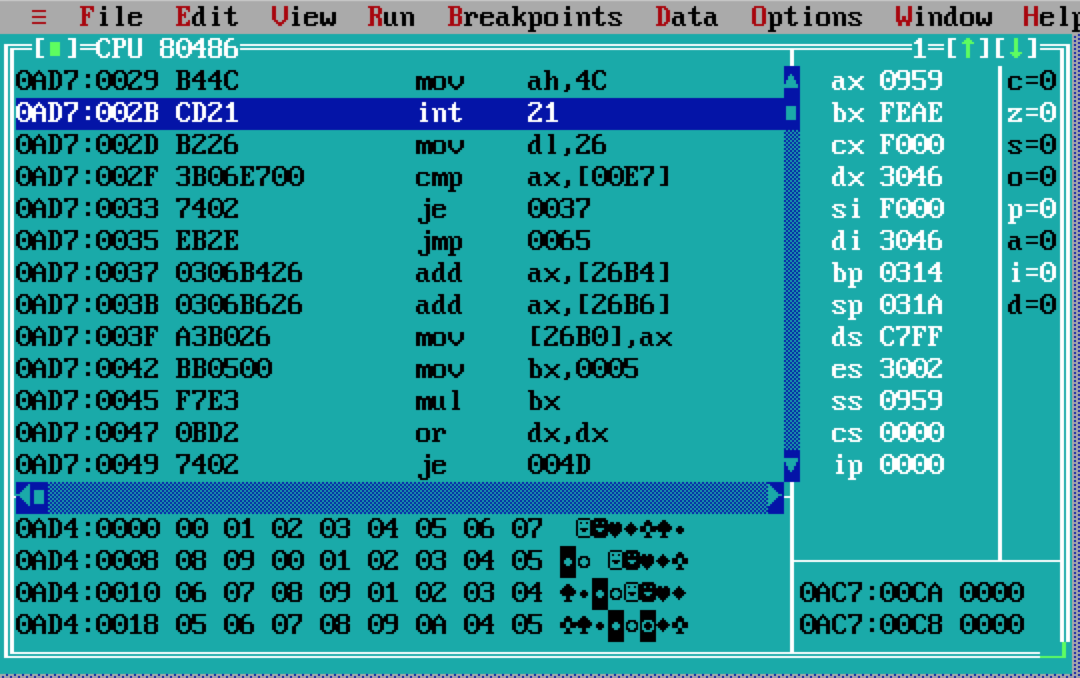


图1.3运行到“INT 21H”时程序运行效果图

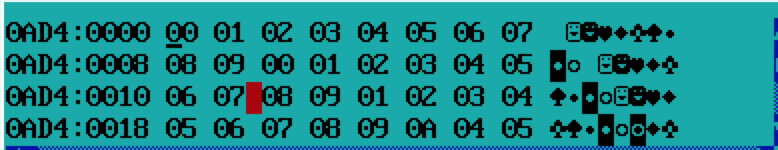


图1.4数据段前40字节内容（1）

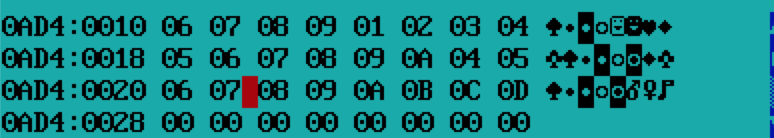


图1.5数据段前40字节内容（2）

3. 源程序意欲将buf1、buf2、buf3、buf4的数据段起始地址分别赋予si、di、bx、bp寄存器，然后利用累加寄存器al将buf1中的数值赋到di之中，所以di中的数据元素为00，01，02，03，04，05，06，07，08，09。后程序将al都分别加1，然后赋到bx段中，故buf3中数据元素为01，02，03，04，05，06，07，08，09，0A。源程序在此的基础上又将al加上3赋予buf4，所以buf4中的数据元素为04，05，06，07，08，09，0A, 0B, 0C, 0D。验证与程序运行结果相同。

1.3.3 任务3

1.3.3.1 设计思想及存储单元分配

将原16位寄存器换做32位寄存器后，应记得16位寄存器与32位寄存器不得同时出现在寻址中。利用变量buf作为偏移地址的位移量，将32位寄存器esi中依次填入al以及自增后的al的值，然后将esi偏移地址加1直至buf1中的数据全部填入。该程序利用cx作为循环的计数寄存器，当cx等于10的时候退出循环，结束程序。

1.存储单元的分配

在数据段中分别为buf1、buf2、buf3、buf4分配10个字节的空间

2.寄存器的分配

AL：用于接收buf1的字符串并转送给buf2、3、4

ESI：作为变址寻址方式中的寄存器，并以buf n作为位移量

CX：作为计数寄存器，用于为循环计数

1.3.3.2 流程图

1.3.3.3 源程序

;----------------------------

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

;----------------------------

DATA SEGMENT USE16

BUF1 DB 0,1,2,3,4,5,6,7,8,9

BUF2 DB 10 DUP(0)

BUF3 DB 10 DUP(0)

BUF4 DB 10 DUP(0)

DATA ENDS

;------------------------------

CODE SEGMENT USE16

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

BEGIN: MOV AX,DATA

MOV DS,AX

MOV ESI,0

MOV CX,0

LOPA: MOV AL,BUF1[ESI]

MOV BUF2[ESI],AL

INC AL

MOV BUF3[ESI],AL

ADD AL,3

MOV BUF4[ESI],AL

INC ESI

INC CX

CMP CX,10

JNE LOPA

MOV AH,4CH ;EXIT

INT 21H

;-----------------------------

CODE ENDS

END BEGIN

1.3.3.4 程序运行效果

1 实验环境条件：2.9GHz Intel Core i5 , 8G内存，macOS Sirra下的DOSbox 0.74; TD.exe6.0

2.buf变量作为esi的位移量，确定了每次向数据段输入数据时的相对位移量，以esi作为变址寄存器使得指针遍历buf数据段中所有的数据。最后利用cx结束计数循环，退出程序。程序运行结果如图3.6所示。对比反汇编代码（[ESI+0000000A],AL）于手打的汇编代码(BUF3[ESI],AL)，发现在反汇编代码中，变量buf n都被转化成了16进制的偏移量，并且buf n与buf n-1之间的偏移量相差正好是10如图3.6与3.7所示。

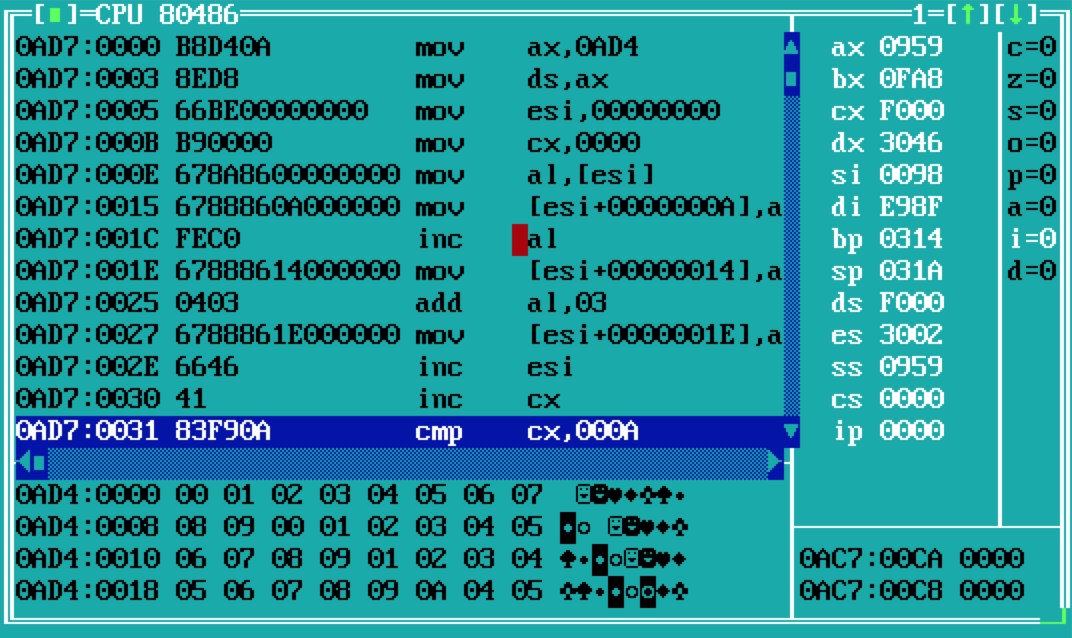


图1.6反汇编码与汇编代码对比（1）

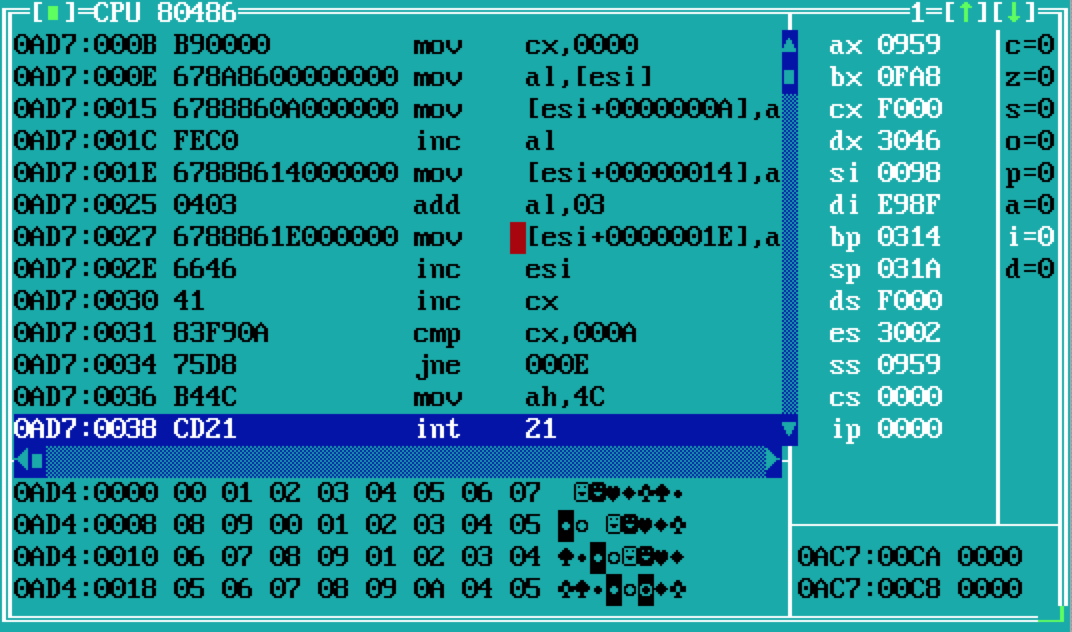


图1.7反汇编代码与汇编代码对比（2）

3.其数据段前40位与任务二相同，如图3.8、3.9所示，与预想相同。

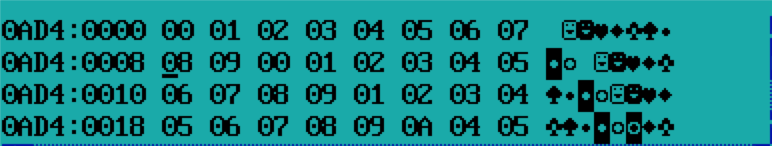


图1.8数据段前40位（1）

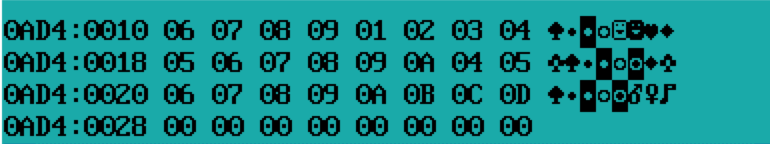


图1.9数据段前40位（2）

4.将ip地址改变后，可以见到代码段直接跳到了未知的内存区域，进行一些无法预料的操作，如图3.10所示，可见ip地址万万不能乱改。

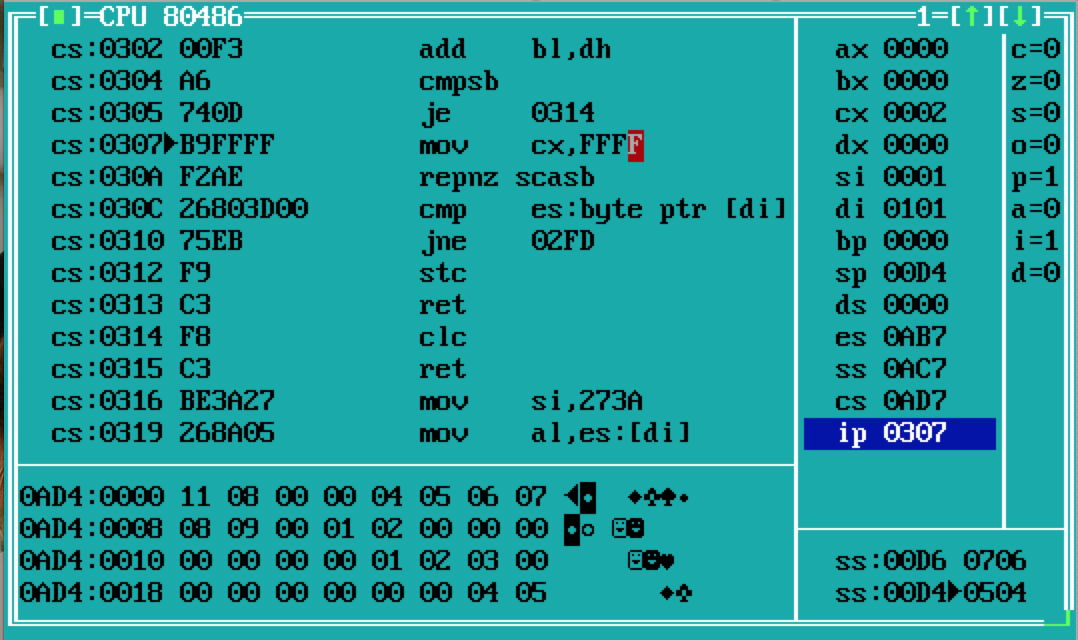


图1.10修改ip地址之后的代码段

1.3.4 任务4

1.3.4.1 设计思想及存储单元分配

与任务三方法类似，利用si以及bp分别标记两数组msg1与msg2以实现将msg1的第一个字符传送到msg2的最后一位字符,并利用si计数，当si等于msg1的长度时退出循环。

1存储单元分配

在数据段中分别为msg1和msg2分配10个字节

2寄存器分配

SI: 用于作为msg1的变址寄存器

BP:用于作为msg2的变址寄存器

1.3.4.2 源代码

;----------------------------

.386

STACK SEGMENT STACK USE16

DB 200 DUP(0)

STACK ENDS

;----------------------------

DATA SEGMENT USE16

MSG1 DB 'Hello 123'

LEN = $-MSG1

MSG2 DB LEN DUP(0)

DATA ENDS

;------------------------------

CODE SEGMENT USE16

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

BEGIN: MOV AX,DATA

MOV DS,AX

MOV SI,0

MOV BP,LEN

LP: MOV AL,MSG1[SI]

MOV MSG2[BP],AL

DEC BP

INC SI

CMP SI,LEN

JNE LP

MOV AH,4CH ;EXIT

INT 21H

;-----------------------------

CODE ENDS

END BEGIN

1.3.4.3 程序运行结果

程序运行结果如图3.10所示，msg2中的字符串为msg1中的字符串的逆序排列，与题意相符。

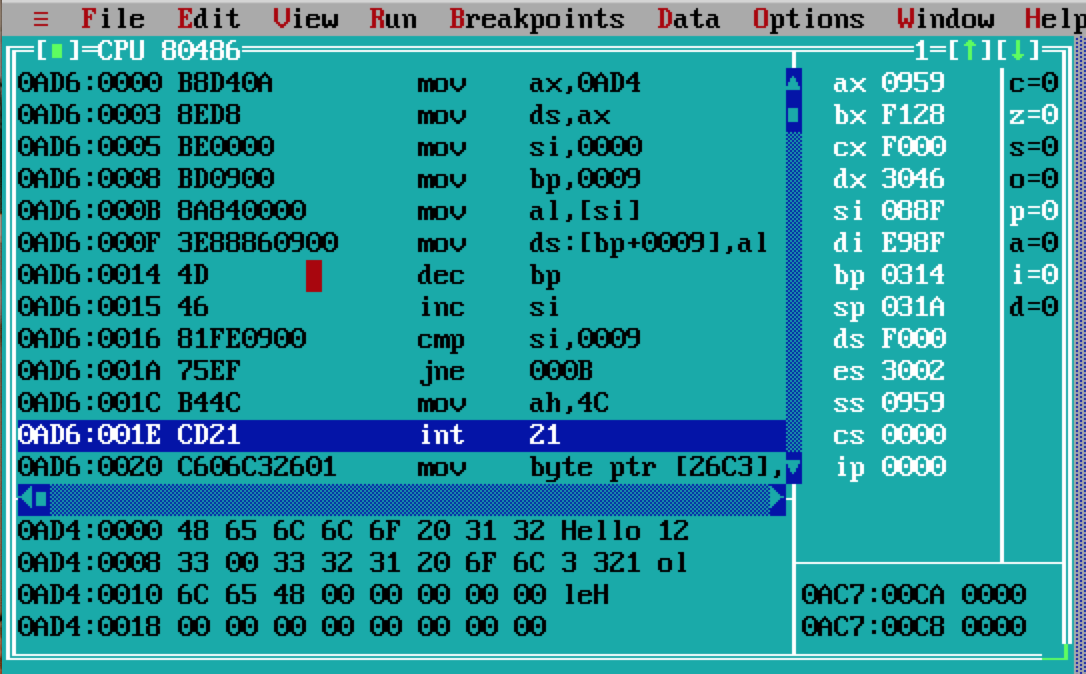


图1.11程序运行结果

# 1.4 实验小结

第一次上机实验，没做什么准备，所以前一个小时都在那里晕晕乎乎的，要不就是不会编译，要不就是找不到编译的程序，总之就是对汇编程序及其不熟练。课前也只是看了一眼实验题目没有深入地思考，导致上手的时候十分手足无措。其次对书本上的内容没有及时地复习，导致我有许多关于编译器使用的错误，在课上也浪费了我好多的时间。虽然前期有那么多的错误，但是感觉汇编写着写着慢慢地有了一点感觉，没刚开始的时候那么迷茫了，就像朱老师所说的：汇编不是看会的，而是做会的。我觉得我还需要对汇编多加练习，要争取做到熟能生巧。

# 1.5参考文献

[1] 王元珍、曹忠升、韩宗芬.80X86汇编语言程序设计(第一版).华中科技大学出版社，2005年:1-348

[2] 许向阳.80X86汇编语言程序设计上机指南(第一版).华中科技大学出版社,2007年:1-224