新世纪高等学校教材

汇编语言程序设计实验指导及习题集

吴向军　李　磊　梁志勇

高等教育出版社
http: // www.hep.com.cn
http: // www.hep.edu.cn

内容提要

本书是与高等教育出版社出版的《汇编语言程序设计》（吴向军等编写）一书相配套的学习辅导书。内容由两部分组成：第一部分是20个实验题，实验内容与教材内容基本上一致，从最基本的调试工具开始，由浅到深、循序渐进来安排实验题，这些实验题在中山大学计算机科学系本科生教学过程中进行了试用，收到了良好的效果。第二部分是教材中大部分练习的参考解答，供学生学习时参考。

在本书的实验部分，还介绍了MASM V6.x编程环境中集成工具——PWB的使用方法，该工具不仅与TC相似，简单易学，而且还有在线帮助，包括：汇编指令的指令格式和功能、ASCII码表、各类按键编码和各种中断的功能等。此外，还介绍了其他编程工具，这些工具允许学生自行设置编程环境，从而使学生真正了解到编程环境的各个方面。

本书可作为高等学校计算机及相关学科的本、专科生学习汇编语言程序设计课程时的辅助资料，也可供应用开发人员参考使用。

图书在版编目(CIP)数据

汇编语言程序设计实验指导及习题集/ 吴向军，李磊，梁志勇. —北京：高等教育出版社，2004.6(2005 重印)

ISBN 7-04-014237-6

Ⅰ. 汇… Ⅱ. ①吴… ②李… ③梁… Ⅲ. 汇编语言-程序设计-高等学校：技术学校-教学参考资料

Ⅳ. TP313

中国版本图书馆 CIP 数据核字（2004）第 042838 号

策划编辑 董建波 **责任编辑** 董建波 **封面设计** 王凌波
版式设计 胡志萍 **责任校对** 杨雪莲 **责任印制** 孔 源

出版发行	高等教育出版社	**购书热线**	010 - 58581118
社 址	北京市西城区德外大街 4 号	**免费咨询**	800 - 810 - 0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010 - 58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经 销	北京蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印 刷	北京明月印务有限责任公司		
开 本	787 × 1092 1/16	**版 次**	2004 年 6 月第 1 版
印 张	15.5	**印 次**	2005 年 11 月第 3 次印刷
字 数	320 000	**定 价**	19.50 元

物料号 14237 - A0

前　言

汇编语言程序设计是计算机专业中一门重要的基础课程，是培养学生直接使用计算机硬件资源能力的一门课程。它不仅能帮助学生进一步理解计算机组成原理中的各种概念，而且还为其他课程，如：操作系统、接口与通信技术和计算机控制技术等，提供必要的预备知识。

汇编语言是与机器相关的编程语言，用它编写的程序可直接利用 CPU 中的资源，也可直接控制各种端口，或利用中断来控制各种硬件资源，程序的执行效率高。因此，有些系统软件的内核和应用程序的核心部分仍用汇编语言来编写。另外，在工业控制方面，汇编语言的应用也很多，所以，通过实习掌握汇编语言的编程和调试方法对将来的应用有极大的帮助。

本书是与高等教育出版社出版的《汇编语言程序设计》(吴向军等编写)相配套的辅助用书，对该教材中的大部分练习给出了一种参考解答，供学生学习时参考，希望起到“抛砖引玉”的作用。

学习汇编语言的一个重要环节就是上机实验，通过实验能对汇编指令的执行效果有一个更深刻的理解，也能对数据在内存的存储形式和变化有一个更清晰的认识。

在本书的实验部分，我们介绍了 MASM V6.x 编程环境下集成工具——PWB 的使用方法，该工具不仅与 TC 相似，简单易学，而且还有在线帮助，包括：汇编指令的指令格式和功能、ASCII 码表、各类按键编码和各种中断的功能等。此外，还介绍了其他编程工具，这些工具允许学生自行设置编程环境，从而使学生真正了解到编程环境的各个方面。

实验部分包括 20 道实验题，实验内容与教材内容的顺序基本上一致，从最基本的调试工具开始，“由浅到深、循序渐进”来安排实验题。这些实验题在中山大学计算机科学系的本科生教学过程中进行了试用，收到了良好的效果。

编者结合多年的教学实践和汇编语言的特点，建议：实验时间要与课堂讲授时间相当，在条件允许的情况下，二者时间之比也可超过 1:1，部分实验题也可安排学生在课余时间自行完成。

在本书编写安排上，吴向军副教授负责整体内容的构思和组织，李磊工程师和梁志勇助工对练习和实验做了大量细致的工作。本书的最后定稿由吴向军副教授负责。

在本书编写过程中，李宏新副教授审阅了全部内容，并结合其多年的教学实践经验，对实验内容和实验安排提出了宝贵的建议，在此，向他表示衷心的感谢。

由于编者能力所限，加之编程工具日新月异的发展，书中不足和谬误之处在所难免，

恳请广大读者批评指正。

E-Mail: issxjwu@zsu.edu.cn，lilei@163.net，isslzy@zsu.edu.cn。

编 者

2004 年 4 月

目　录

第一篇 实验指导

实验一 调试工具 DEBUG 的使用

1.1 实验目的

（1）学习如何在 Windows 的命令模式下启动 DEBUG。

（2）掌握 DEBUG 的常用基本命令。

（3）学习如何用 DEBUG 进行跟踪调试。

1.2 预备知识

1．进制转换

熟练掌握二进制、八进制、十进制和十六进制的相互转换算法。

2．寄存器

寄存器是 CPU 内部的数据存储资源，是汇编程序员能直接使用的硬件资源之一。寄存器的存取速度比 Cache 还要快。

在 16 位 CPU 中，总共有 4 个 16 位数据寄存器 AX、BX、CX 和 DX，每个 16 位寄存器又可分为 2 个 8 位寄存器（例如 AX 的高 8 位称为 AH，低 8 位称为 AL）；2 个变址寄存器 DI 和 SI；2 个指针寄存器 SP 和 BP；4 个段寄存器 ES、CS、SS 和 DS；1 个标志寄存器 FLAGS；1 个指令指针寄存器 IP。

3．标志位

标志寄存器 FLAGS 的每个位都可以作为标志位。16 位 CPU 使用其中 8 个位表示溢出、中断、进位等状态。每个标志位都有置位和复位两种状态，它们在 DEBUG 中的表示如表 1.1 所示。

表 1.1 DEBUG 中标志位的符号表示

标志名称	溢出	方向	中断	负号	零	辅助进位	奇偶	进位
	OF	DF	IF	SF	ZF	AF	PF	CF
置位状态	OV	DN	EI	NG	ZR	AC	PE	CY
复位状态	NV	UP	DI	PL	NZ	NA	PO	NC

1.3 DEBUG 命令集

DEBUG 命令及其含义如表 1.2 所示。

表 1.2 DEBUG 命令及其含义

命 令 格 式		功 能 说 明
A	[地址]	输入汇编指令
C	范围 起始地址	对由“范围”指定的区域与“起始地址”指定的同大小区域进行比较，显示不相同的单元
D	[范围]	显示指定范围内的内存单元内容
E	起始地址 字节值表	用值表中的值替换从“起始地址”开始的内存单元内容
F	范围 字节值表	用指定的字节值表来填充内存区域
G	[=起始地址][断点地址]	从起点（或当前地点）开始执行，到终点结束
H	数值 1 数值 2	显示两个十六进制数值之和、差
I	端口地址	从计算机输入端口读取数据并显示
L	[地址 [驱动器号 扇区 扇区数]]	从磁盘中将文件或扇区内容读入内存
M	范围 地址	把“范围”内的字节值传送到从“地址”开始的单元
N	文件标识符 [文件标识符…]	指定文件名，为读/写文件做准备
O	端口地址 字节值	向计算机输出端口送出数据
P	[=地址][指令数]	按执行过程，但不进入子程序调用或软中断
Q		退出 DEBUG，回到 DOS 状态
R	[寄存器名]	显示和修改寄存器内容
S	范围 字节值表	在内存区域内搜索指定的字节值表。如果找到，显示起始地址，否则，什么也不显示
T	[=地址][指令数]	跟踪执行，从起点（或当前地点）执行若干条指令
U	[范围]	反汇编，显示机器码所对应的汇编指令
W	[地址 [驱动器号 扇区 扇区数]]	向磁盘写内容，（BX、CX）为写入字节数

关于参数的几点说明：

（1）进制：在 DEBUG 中输入或显示的数据都是十六进制形式；

（2）分隔：命令和参数、参数和参数之间要用空格、逗号或制表符等分隔；

（3）地址：用“段值:偏移量”的形式来表示地址，也可用段寄存器来代表“段值”。

例如：1000:0、ds:10、es:200、cs:30 等。

① 范围：表示地址范围，它有两种表示方式：“地址 1 地址 2”和“地址 1 长度”。其中：“地址 1”表示起始地址，要用“段值:偏移量”来表示；“地址 2”表示终止地址，只用“偏移量”来表示；“长度”用字母 L 开头的数值来表示。

例如：100:50 100——段值为 100，偏移量从 50 到 100 的内存区域；

100:50 L100——段值为 100，偏移量从 50 开始的 100 个字节区域。

② 端口地址：2 位十六进制数值；

③ 字节值：2 位十六进制数值；

④ 字节值表：由若干个字节值组成，也可以是用引号括起来的字符串；

⑤ 驱动器号：0 表示驱动器 A、1 表示驱动器 B、2 表示驱动器 C、3 表示驱动器 D 等。

1.4　DEBUG 命令示例

1.4.1　启动 DEBUG

1．打开 Windows 命令窗口

打开命令窗口的步骤：选择“开始”→“运行”。

（1）在 Windows 95/98 环境中，输入“command”命令；

（2）在 Windows 2000 环境中，输入“cmd”命令，如图 1.1 所示。

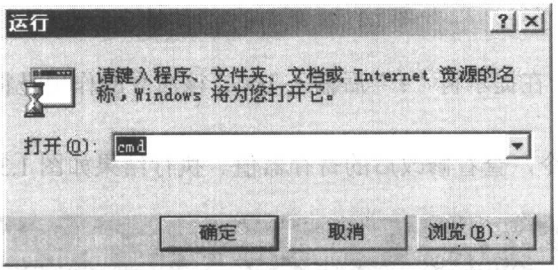

图 1.1　Windows 2000 环境下运行命令窗口

2．启动 DEBUG

在命令窗口中启动 DEBUG，一般命令如下：

DEBUG　[文件名] [参数表]

其中：文件名指定被调试的文件，其包括名称和后缀，参数表是被调试文件运行时所需要的参数。被调试的文件可以是系统中的任何文件，但通常它们的后缀为.EXE 或.COM。

当 DEBUG 启动成功后，将显示连接符“-”，这时，可输入各种 DEBUG 命令。DEBUG 中标志位的符号表示如表 1.1 所示，其所有命令及其含义如表 1.2 所示。

关于使用命令的几点说明：

（1）在提示符“-”下才能输入命令，在按“回车”键后，该命令才开始执行；

（2）命令是单个字母，命令和参数的大小写可混合输入；

（3）可用 F1、F2、F3、Ins、Del、←、→等编辑键来编辑本行命令；

（4）当命令出现语法错误时，将在出错位置显示“^Error”；

（5）可用 Ctrl+C 或 Ctrl+Break 来终止当前命令的执行，还可用 Ctrl+S 来暂停屏幕显示（当连续不断地显示信息时）；

（6）在 DEBUG 中使用的数都是以十六进制来表示；

（7）DEBUG 中的命令不区分大小写。

下面对 DEBUG 中的命令逐个介绍，并实现相应的示例来加以说明。由于不同计算机环境的差异，下面所给的输出结果在不同计算机中可能有所不同。

1.4.2 R 命令

R 命令作用：显示和修改寄存器的值和标志位的状态。

输入命令 R，将显示所有寄存器的值和标志位的状态。执行结果如图 1.2 所示。

```
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B07  ES=0B07  SS=0B07  CS=0B07  IP=0100  NV UP EI PL NZ NA PO NC
0B07:0100 1E            PUSH    DS
-_
```

图 1.2 R 命令的执行结果

输入命令 R CX，在提示符“:”后输入 100。该命令的作用是将寄存器 CX 的值设置为 100。

再输入执行 R 命令，查看修改后的寄存器值。执行结果如图 1.3 所示。

```
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B07  ES=0B07  SS=0B07  CS=0B07  IP=0100  NV UP EI PL NZ NA PO NC
0B07:0100 1E            PUSH    DS
-R CX
CX 0000
:100
-R
AX=0000  BX=0000  CX=0100  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B07  ES=0B07  SS=0B07  CS=0B07  IP=0100  NV UP EI PL NZ NA PO NC
0B07:0100 1E            PUSH    DS
-_
```

图 1.3 R CX 命令的执行结果

1.4.3 H 命令

H 命令作用：计算两个十六进制数的和与差。

输入命令：H 10 1，执行结果如图 1.4 所示。

```
-H 10 1
0011  000F
-_
```

图 1.4 H 10 1 命令的执行结果

命令的输出结果有两个数，前者是命令中两个参数之和，后者是两个参数之差。

1.4.4　D 命令

D 命令作用：显示指定内存区域的内容。

输入命令 D，执行结果如图 1.5 所示。

```
-D
0B07:0100  1E 99 25 02 02 0A C0 74-13 3A C4 75 0F 80 3E D9   ..%....t.:.u..>.
0B07:0110  99 00 74 08 FE 06 17 99-32 C0 EB 06 34 00 F6 0A   ..t.....2...4...
0B07:0120  D0 E8 0A 06 D9 99 A2 C7-96 D0 E0 D0 E0 A2 D2 99   ................
0B07:0130  80 3E D4 99 00 75 24 A2-D8 99 0A C9 75 1D 0A C0   .>...u$.....u...
0B07:0140  74 19 8B 0E D5 96 E3 13-B0 1A 06 33 FF 8E 06 B4   t..........3....
0B07:0150  96 F2 AE 07 75 05 4F 89-3E D5 96 BB BA 97 80 3E   ....u.O.>......>
0B07:0160  C7 96 00 74 03 BB 00 98-BE 77 97 8B 3E B9 98 B9   ...t.....w..>...
0B07:0170  08 00 E8 12 00 80 3C 20-74 09 B0 2E AA B9 03 00   ......< t.......
-
```

图 1.5　D 命令的执行结果

说明：（1）XXXX:YYYY —— 前者是内存单元的段地址，后者是内存单元的偏移量；

（2）中间显示区域是内存单元内容（十六进制的形式），每行显示 16 个字节的内容；

（3）右边以“字符”形式显示内存单元值。

命令 D 可带参数，也可省略参数。省略参数时，第一次命令 D 显示的内容是从 DS:100 开始的 128 个字节，再执行不带参数的命令 D 时，则从上次的显示位置往下接着显示。

若执行带参数的命令 D，DEBUG 将显示指定地址范围的内容。带参数的方式有三种。

方式一：D 起始位置

DEBUG 从起始位置开始显示 128 个字节的内容。输入命令 D 1AF5:100，执行结果如图 1.6 所示。

```
-D1AF5:100
1AF5:0100  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0110  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
```

图 1.6　D 1AF5:100 命令的执行结果

方式二：D 起始位置 结束位置。

DEBUG 从起始位置开始一直显示到结束位置。输入命令 D DS:100 11F，执行结果如图 1.7 所示。

```
-D DS:100 11F
0B07:0100  1E 99 25 02 02 0A C0 74-13 3A C4 75 0F 80 3E D9   ..%....t.:.u..>.
0B07:0110  99 00 74 08 FE 06 17 99-32 C0 EB 06 34 00 F6 0A   ..t.....2...4...
-
```

图 1.7　D DS:100 11F 命令的执行结果

方式三：D 起始位置 L 长度

DEBUG 从起始位置开始显示指定长度的内容。输入命令 D DS:100 L10，执行结果如图 1.8 所示。

```
-D DS:100 L10
0B07:0100  1E 99 25 02 02 0A C0 74-13 3A C4 75 0F 80 3E D9   ..%....t.:.u..>.
-
```

图 1.8 D DS:100 L10 命令的执行结果

1.4.5 E 命令

E 命令作用：用字节值表中的值替换从“起始地址”开始的内存单元内容。

E 命令的使用方式：E 起始地址 字节值表

E 命令的使用方式也可以是：E 起始地址。

输入命令：E 1AF5:0100，连续输入 12（空格）、34（空格）、（空格）、56（回车）；再输入命令 D 1AF5:0100，执行结果如图 1.9 所示。

```
-E 1AF5:0100
1AF5:0100  00.12   00.34   00.     00.56
-D 1AF5:0100
1AF5:0100  12 34 00 56 00 00 00 00-00 00 00 00 00 00 00 00   .4.V............
1AF5:0110  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
```

图 1.9 E 1AF5:0100 命令的执行结果

DEBUG 显示 1AF5:0100 单元的内容 12，这时可输入新的数值以达到修改该字节值的目的。如果还要修改后续字节的内容，可按“空格键”继续。当要跳过某个字节时，可以连续按两个空格。

1.4.6 F 命令

F 命令作用：用指定的字节值表来填充内存区域。

F 命令的使用方式：F 范围 字节值表

输入命令：F 1AF5:100 L20 1 2 3 4 5，再输入命令 D 1AF5:100，执行结果如图 1.10 所示。

```
-F 1AF5:100 L20 1 2 3 4 5
-D 1AF5:100
1AF5:0100  01 02 03 04 05 01 02 03-04 05 01 02 03 04 05 01   ................
1AF5:0110  02 03 04 05 01 02 03 04-05 01 02 03 04 05 01 02   ................
1AF5:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
```

图 1.10 F 1AF5:100 L20 1 2 3 4 5 命令的执行结果

说明：该命令是用数值序列 01、02、03、04、05 依次填充到从 1AF5:100 开始长度为 20H 的内存区域。

输入命令：F 1AF5:100 13F 41 42 43 44，再执行 D 1AF5:100 显示内存单元的内容，执行结果如图 1.11 所示。

```
-F 1AF5:100 13F 41 42 43 44
-D 1AF5:100
1AF5:0100  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0110  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0120  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0130  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1AF5:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
```

图 1.11　F 1AF5:100 13F 41 42 43 44 命令的执行结果

说明：该命令是用数值序列 41、42、43、44 循环填充到 1AF5:[100 ~ 13F]的内存区域。

1.4.7　M 命令

M 命令作用：把“范围”内的字节值传送到从“地址”开始的单元。

M 命令的使用方式：M 范围 地址

输入命令：M 1AF5:100 13F 1AF5:140，再输入命令 D 1AF5:100 显示内存单元的内容，执行结果如图 1.12 所示。

```
-M 1AF5:100 13F 1AF5:140
-D 1AF5:100
1AF5:0100  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0110  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0120  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0130  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0140  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0150  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0160  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0170  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
-
```

图 1.12　M 1AF5:100 13F 1AF5:140 命令的执行结果

说明：把内存单元 1AF5:[100 ~ 13F]的内容复制到从 1AF5:140 开始的内存单元。

1.4.8　C 命令

C 命令作用：对由“范围”指定的区域与“起始地址”指定的同大小区域进行比较，显示不相同的单元。

C 命令的使用方式：C 范围 起始地址

说明：将指定范围的内存区域与从指定地址开始的相同长度的内存区域逐个字节进行比较，列出不同的内容。

输入命令：C 1AF5:100 13F 1AF5:140。由于两块内容完全相同，所以命令执行后没有

任何显示。再输入命令：C 1AF5:100 107 1AF5:180，命令执行后列出比较结果不同的各个字节。命令执行结果如图 1.13 所示。

```
-C 1AF5:100 13F 1AF5:140
-C 1AF5:100 107 1AF5:180
1AF5:0100  41  00  1AF5:0180
1AF5:0101  42  00  1AF5:0181
1AF5:0102  43  00  1AF5:0182
1AF5:0103  44  00  1AF5:0183
1AF5:0104  41  00  1AF5:0184
1AF5:0105  42  00  1AF5:0185
1AF5:0106  43  00  1AF5:0186
1AF5:0107  44  00  1AF5:0187
-
```

图 1.13 C 命令的执行结果

1.4.9 S 命令

S 命令作用：在内存区域内搜索指定的字节值表。如果找到，显示起始地址，否则，什么也不显示。

S 命令的使用方式：S 范围 字节值表

输入命令：D 1AF5:100 11F，显示该区域的内存单元内容。

输入命令：S 1AF5:100 11F 41 42 43 44。搜索该区域是否存在字节串 41 42 43 44，并将搜索结果输出。执行结果如图 1.14 所示。

```
-D 1AF5:100 11F
1AF5:0100  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
1AF5:0110  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44   ABCDABCDABCDABCD
-S 1AF5:100 11F 41 42 43 44
1AF5:0100
1AF5:0104
1AF5:0108
1AF5:010C
1AF5:0110
1AF5:0114
1AF5:0118
1AF5:011C
-
```

图 1.14 S 1AF5:100 11F 41 42 43 44 命令的执行结果

从上面的执行结果可以看出，总共搜索出 8 处匹配。

1.4.10 A 命令

A 命令作用：输入汇编指令。

以下的程序要在屏幕上显示“ABCD”四个字符。

先用 E CS:200 命令将“ABCD$”四个字符预先放在内存 CS:200 处(“ABCD$”的 ASCII 码的十六进制数分别为：41、42、43、44、24)，然后用命令 D CS:200 20F 显示内存数据内容，最后用命令 A 100 输入以下汇编程序指令：

```
XXXX:0100 MOV AX,CS
XXXX:0102 MOV DS,AX
XXXX:0104 MOV DX,200
XXXX:0107 MOV AH,9
XXXX:0109 INT 21
XXXX:010B INT 20
```

执行结果如图 1.15 所示。

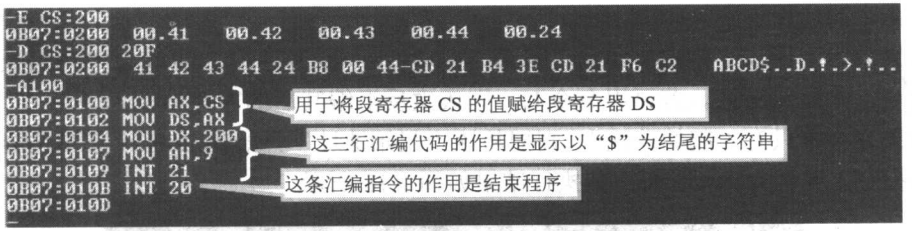

图 1.15　E CS:200 和 A100 命令的执行结果

1.4.11　G 命令

G 命令作用：从起点（或当前地点）开始执行汇编指令，到终点结束。

G 命令的使用方式：G [=起始地址] [断点地址]

其作用是从起始地址开始执行到断点地址。如果不设置断点，则程序一直运行到终止指令才停止。

在完成上节的内存数据并输完相应的程序后，在 DEBUG 中执行命令 G=100，输出结果如图 1.16 所示。

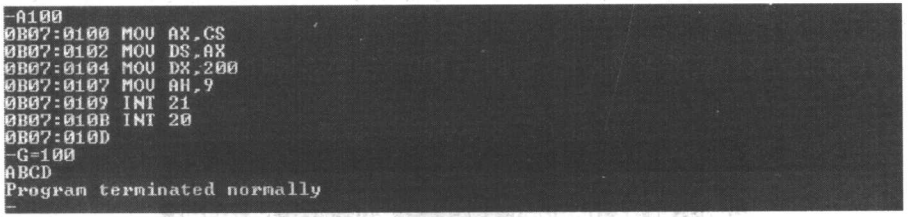

图 1.16　G=100 命令的执行结果

汇编程序运行后在屏幕上显示出“ABCD”4 个字符。

在 DEBUG 中，再执行 G =100 10B，其含义是从地址 CS:100 开始，一直运行到 CS:10B 停止。其执行结果如图 1.17 所示。

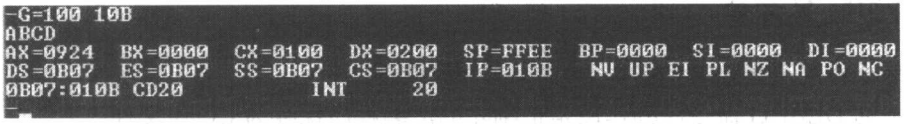

图 1.17　G=100 10B 命令的执行结果

命令执行后，不但显示出字符串“ABCD”，而且还列出当前寄存器和标志位的值。

1.4.12　U 命令

U 命令作用：反汇编，显示机器码所对应的汇编指令。

U 命令的使用方式：U [范围]

如果命令中不写“范围”参数，那么，把从 CS:100 开始的 20H 个字节内容反汇编成汇编指令，并显示出来。若再用命令 U，则依次反汇编其后的内存单元。

如果命令中只有“起始地址”，则把从该“起始地址”开始的 20H 个字节内容反汇编成汇编指令。执行命令 U 100，其执行结果如图 1.18 所示。

```
-U 100
0B07:0100 8CC8          MOV     AX,CS
0B07:0102 8ED8          MOV     DS,AX
0B07:0104 BA0002        MOV     DX,0200
0B07:0107 B409          MOV     AH,09
0B07:0109 CD21          INT     21
0B07:010B CD20          INT     20
0B07:010D 803ED99900    CMP     BYTE PTR [99D9],00
0B07:0112 7408          JZ      011C
0B07:0114 FE061799      INC     BYTE PTR [9917]
0B07:0118 32C0          XOR     AL,AL
0B07:011A EB06          JMP     0122
0B07:011C 3400          XOR     AL,00
0B07:011E F60A          ???     BYTE PTR [BP+SI]
-
```

图 1.18 U 100 命令的执行结果

如果命令中只有“起始地址”和“终止地址”，则把从“起始地址”到“终止地址”之间的字节内容反汇编成汇编指令。

命令 U 100 10C 的作用是对从 CS:100 到 10B 的内存单元进行反汇编，其执行结果如图 1.19 所示。

```
-U 100 10C
0B07:0100 8CC8          MOV     AX,CS
0B07:0102 8ED8          MOV     DS,AX
0B07:0104 BA0002        MOV     DX,0200
0B07:0107 B409          MOV     AH,09
0B07:0109 CD21          INT     21
0B07:010B CD20          INT     20
-
```

图 1.19 U 100 10C 命令的执行结果

1.4.13 N 命令

N 命令作用：指定文件名，为读/写文件做准备。

以下的 DEBUG 命令序列作用是把刚才的汇编程序保存为 COM 文件。

```
D 200 20F
U 100 10C
N E:\FIRST.COM
R CX
:110
W
```

第一、二条命令的作用是检查一下刚才编写的汇编指令，第三条命令的作用是设置存盘文件名为 FIRST.COM，第四条命令的作用是设置存盘文件大小为 110H 个字节。最后一条命令是将文件存盘。执行结果如图 1.20 所示。

```
-D 200 20F
0B07:0200  41 42 43 44 24 B8 00 44-CD 21 B4 3E CD 21 F6 C2   ABCD$..D.!.>.!..
-U 100 10C
0B07:0100 8CC8          MOV     AX,CS
0B07:0102 8ED8          MOV     DS,AX
0B07:0104 BA0002        MOV     DX,0200
0B07:0107 B409          MOV     AH,09
0B07:0109 CD21          INT     21
0B07:010B CD20          INT     20
-N E:\FIRST.COM
-R CX
CX 0100
:110
-W
Writing 00110 bytes
```

图 1.20　N E:\FIRST.COM 命令的执行结果

文件存盘后，在 DOS 环境下执行 FIRST.COM，观看存盘的可执行文件的运行结果，如图 1.21 所示。

```
E:\>FIRST
ABCD
E:\>
```

图 1.21　FIRST.COM 文件的运行结果

1.4.14　W 命令

W 命令作用：向磁盘写内容。

在上节“N 命令”中可以看出如何使用 W 命令将文件存盘。

在没有很好地掌握汇编语言和磁盘文件系统前，暂时不要使用 W 命令写磁盘扇区，否则很容易损坏磁盘文件，甚至破坏整个磁盘的文件系统。

1.4.15　L 命令

L 命令作用：从磁盘中将文件或扇区内容读入内存。

将文件调入内存必须先用 DEBUG 的 N 命令设定文件名。下面的命令是将 FIRST.COM 读入内存。

```
N E:\FIRST.COM
L
```

在文件读入内存后，可用 U 100 命令查看调入程序的汇编代码。执行结果如图 1.22 所示。

```
-N E:\FIRST.COM
-L
-U 100
0B54:0100 8CC8          MOV     AX,CS
0B54:0102 8ED8          MOV     DS,AX
0B54:0104 BA0002        MOV     DX,0200
0B54:0107 B409          MOV     AH,09
0B54:0109 CD21          INT     21
0B54:010B CD20          INT     20
0B54:010D 803ED99900    CMP     BYTE PTR [99D9],00
0B54:0112 7408          JZ      011C
0B54:0114 FE061799      INC     BYTE PTR [9917]
0B54:0118 32C0          XOR     AL,AL
0B54:011A EB06          JMP     0122
0B54:011C 3400          XOR     AL,00
0B54:011E F60A          ???     BYTE PTR [BP+SI]
-
```

图 1.22 L 命令的执行结果

读取磁盘扇区的方式：L [内存地址[磁盘驱动器号 起始扇区 扇区数]]

内存地址：指定存放读入内容（加载文件或扇区内容）的内存起址，若不指定“内存地址”，则 DEBUG 将使用 CS:100 为当前地址。

磁盘驱动器号：指定包含读取指定扇区的磁盘的驱动器，其值可为：0=A，1=B，2=C 等。

起始扇区：指定要加载其内容的第一个扇区的十六进制数。“扇区数”指定要加载其内容的连续扇区的十六进制数。

只有要加载特定扇区的内容时，才能使用[磁盘驱动器号] [起始扇区] [扇区数]参数。

例如：要将 C 盘第一扇区读取到内存 DS:300 的位置，相应的 DEBUG 命令为 L DS:300 2 1 1。但是由于 Windows 操作系统对文件系统的保护，这条命令可能会被操作系统禁止运行，如图 1.23 所示。

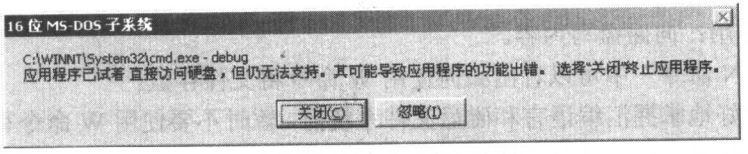

图 1.23 Windows 的错误提示

1.4.16 T 命令

T 命令作用：跟踪执行，从起点（或当前点）执行若干条指令。

T 命令的使用方式：T [=地址] [指令数]

若不写“地址”，则 T 命令从 CS：IP 处开始执行，“指令数”是要执行的指令数。若只用 T 命令，则从 CS:IP 处执行一条指令。

执行以下 6 条命令：

```
N E:\FIRST.COM
L
```