

高等院校计算机实验与实践系列示范教材

汇编语言实验教程

张 坤 编著

清华大学出版社



TP313/114

2008

高等院校计算机实验与实践系列示范教材

汇编语言实验教程

张 坤 编著

清华大学出版社

北京

内 容 简 介

本书是“汇编语言程序设计”课程的配套实验教材。全书选用多个具有代表性的实验,对汇编语言结构化和模块化程序进行了深入解析,详细叙述了汇编语言程序的编程与调试过程。

本书以 Intel 80x86 指令系统为主体,在 MS-DOS 和 Windows 环境下较为系统地介绍了汇编语言指令的用法和调试工具 DEBUG、W32Dasm 的使用。

第 1 章介绍了汇编语言上机过程和 DEBUG 的使用;第 2 章详细介绍了汇编指令的用法;第 3 章介绍了伪指令的使用;第 4 章~第 7 章重点介绍了顺序、分支、循环、子程序程序设计的实例;第 8 章介绍了 Win32 汇编程序设计方法;附录部分给出了自选练习题及答案。

本书是高等院校计算机、自动化、电子技术及相关专业“汇编语言程序设计”课程的实验教材,还可以作为汇编语言程序设计的自学教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

汇编语言实验教程/张坤编著. —北京:清华大学出版社,2008.7

(高等院校计算机实验与实践系列示范教材)

ISBN 978-7-302-17716-6

I. 汇… II. 张… III. 汇编语言—程序设计—高等学校—教材 IV. TP313

中国版本图书馆 CIP 数据核字(2008)第 075209 号

责任编辑:索 梅

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:17.75 字 数:424 千字

版 次:2008 年 7 月第 1 版 印 次:2008 年 7 月第 1 次印刷

印 数:1~4000

定 价:27.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028650-01

出版说明

当前,重视实验与实践教育是各国高等教育界的发展潮流,我国与国外教学工作的差距也主要表现在实践教学环节上。面对新的形式和新的挑战,完善实验与实践教育体系成为一种必然。为了培养具有高质量、高素质、高实践能力和高创新能力的人才,全国很多高等院校在实验与实践教学方面进行了大力改革,在实验与实践教学内容、教学方法、教学体系、实验室建设等方面积累了大量的宝贵经验,起到了教学示范作用。

实验与实践性教学与理论教学是相辅相成的,具有同等重要的地位。它是在开放教育的基础上,为配合理论教学、培养学生分析问题和解决问题的能力以及加强训练学生专业实践能力而设置的教学环节;对于完成教学计划、落实教学大纲,确保教学质量,培养学生分析问题、解决问题的能力和实际操作技能更具有特别重要的意义。同时,实践教学也是培养应用型人才的重要途径,实践教学质量的好坏,实际上也决定了应用人才培养质量的高低。因此,加强实践教学环节,提高实践教学质量,对培养高质量的应用型人才至关重要。

近年来,教育部把实验与实践教学作为对高等院校教学工作评估的关键性指标。2005年1月,在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出:“高等学校要强化实践育人的意识,区别不同学科对实践教学的要求,合理制定实践教学方案,完善实践教学体系。要切实加强实验、实习、社会实践、毕业设计(论文)等实践教学环节,保障各环节的时间和效果,不得降低要求。”“要不断改革实践教学内容,改进实践教学方法,通过政策引导,吸引高水平教师从事实践环节教学工作。要加强产学研合作教育,充分利用国内外资源,不断拓展校际之间、校企之间、高校与科研院所之间的合作,加强各种形式的实践教学基地和实验室建设。”

为了配合开展实践教学及适应教学改革的需要,我们在全中国各高等院校精心挖掘和遴选了一批在计算机实验与实践教学方面具有潜心研究并取得了富有特色、值得推广的教学成果的作者,把他们多年积累的教学经验编写成教材,为开展实践教学的学校起一个抛砖引玉的示范作用。

为了保证出版质量,本套教材中的每本书都经过编委会委员的精心筛选和严格评审,坚持宁缺毋滥的原则,力争把每本书都做成精品。同时,为了能够让更多、更好的实践教学经验应用于社会和各高等院校,我们热切期望在这方面有经验和成果的教师能够加入到本套丛书的编写队伍中,为实践教学的发展和取得实效做出贡献;也衷心地期望广大读者对本套教材提出宝贵意见,以便我们更好地为读者服务。

清华大学出版社

联系人:索梅 suom@tup.tsinghua.edu.cn

在目前正在使用的程序设计语言中,汇编语言是被误解得最深的一种。当提到“汇编语言”这个术语时,经常使人联想到低级的位移动和复杂难记的指令格式。随着各种出色的高级语言开发工具的快速发展,“汇编语言程序设计已经死了”这种评论并不少见。但是汇编语言程序设计还远没有到死亡的时候,每种高级语言程序在连接为可执行程序之前都必须被编译为汇编语言程序。

概括地说,如果读者想从事计算机科学方面的工作,汇编语言的基础是必不可缺的。原因很简单,我们的工作平台、研究对象都是机器,汇编语言是人和计算机沟通最直接的方式,它描述了机器最终所要执行的指令序列。汇编语言与具体的微处理器相关,每种微处理器的汇编语言都不一样,我们只能通过一种常用的、结构简洁的微处理器的汇编语言来进行学习,从而达到学习汇编语言的两个根本的目的:充分获得底层编程的体验,深刻理解机器运行程序的机理。

目前,汇编语言更多地使用于 Windows 环境下,例如病毒分析、软件调试和软件加密。学习完 Windows 下的汇编语言,对面向对象程序设计会有较为深入的理解,但是 DOS 下的汇编语言程序设计还是基础。

本书通过一系列使用 80x86 和 Windows 环境下 32 位汇编语言的实验练习,把汇编语言程序设计的概念和理论知识融入实践中,从而加深对汇编语言程序设计的认识和理解。每个实验中包含背景知识介绍和按步骤进行的实验指导等,实验内容的组织充分顾及了不同的难易程度,富有挑战性。本书具有很好的可读性和可操作性,实用性比较强,结构清晰,着重培养动手能力。

本书附录部分给出了大量各种类型的习题及答案,为学生课余时间自学和参加等级考试提供了有力的帮助。

本书配有电子课件,读者可从清华大学出版社网站(<http://www.tup.com.cn>)下载。

由于编者水平有限,错误和不妥之处,敬请读者提出宝贵意见。

张 坤
2008 年 6 月

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收

邮编：100084

电子邮件：jsjic@tup.tsinghua.edu.cn

电话：010-62770175-4608/4409

邮购电话：010-62786544

教材名称：汇编语言实验教程

ISBN 978-7-302-17716-6

个人资料

姓名：_____ 年龄：_____ 所在院校/专业：_____

文化程度：_____ 通信地址：_____

联系电话：_____ 电子信箱：_____

您使用本书是作为：☐指定教材 ☐选用教材 ☐辅导教材 ☐自学教材

您对本书封面设计的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书印刷质量的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书的总体满意度：

从语言质量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

从科技含量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

本书最令您满意的是：

☐指导明确 ☐内容充实 ☐讲解详尽 ☐实例丰富

您认为本书在哪些地方应进行修改？（可附页）

您希望本书在哪些方面进行改进？（可附页）

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案(素材)，有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案(素材)，希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjic@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页 (<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>) 上查询。

CONTENTS

目录

第 1 章 汇编语言上机过程和 DEBUG 的使用	1
1.1 实验目的	1
1.2 预备知识	1
1.2.1 汇编语言程序的建立和执行	1
1.2.2 调试程序 DEBUG	4
1.3 实验内容	16
1.3.1 汇编语言上机过程	16
1.3.2 使用 DEBUG 调试程序	17
1.4 实验报告要求	19
第 2 章 汇编指令的用法	20
2.1 实验目的	20
2.2 预备知识	20
2.2.1 80x86 寄存器组	20
2.2.2 80x86 的寻址方式	22
2.2.3 8086/8088 指令	23
2.3 实验内容	35
2.3.1 数据传送指令的用法	35
2.3.2 算术指令的用法	36
2.3.3 逻辑运算指令的用法	37
2.3.4 条件转移指令的用法	37
2.3.5 移位指令的用法	38
2.3.6 串指令的用法	38
2.4 实验报告要求	42
第 3 章 伪操作(伪指令)	43
3.1 实验目的	43
3.2 预备知识	43
3.2.1 数据定义伪指令	43

3.2.2	操作数运算符	44
3.2.3	段定义伪指令	49
3.2.4	简化的段定义	51
3.3	实验内容	53
3.3.1	伪指令的用法	53
3.3.2	数据定义及存储器分配伪操作	55
3.3.3	简化的段定义举例	55
3.4	实验报告要求	56
第4章	顺序、分支程序设计	57
4.1	实验目的	57
4.2	预备知识	57
4.3	实验内容	63
4.3.1	顺序程序设计	63
4.3.2	分支程序设计	65
4.4	实验报告要求	71
第5章	循环程序设计	72
5.1	实验目的	72
5.2	预备知识	72
5.2.1	循环程序的组成	72
5.2.2	循环控制方法	72
5.3	实验内容	77
5.4	实验报告要求	80
第6章	子程序设计	81
6.1	实验目的	81
6.2	预备知识	81
6.2.1	子程序的调用和返回	81
6.2.2	宏定义	86
6.2.3	宏指令与子程序的区别	87
6.3	实验内容	87
6.4	实验报告要求	94
第7章	中断程序设计	95
7.1	实验目的	95
7.2	预备知识	95
7.2.1	BIOS 和 DOS 中断	95

7.2.2 BIOS 和 DOS 中断调用应用举例	95
7.3 实验内容	97
7.4 实验报告要求	103
第 8 章 Win32 汇编程序设计	104
8.1 实验目的	104
8.2 预备知识	104
8.2.1 32 位寻址方式	104
8.2.2 80x86 及 Pentium CPU 扩充和增加的指令	107
8.2.3 DOS 32 位程序举例	116
8.2.4 Windows 下 32 位汇编程序	117
8.2.5 MASM32 的使用	122
8.2.6 Windows 程序的反汇编	124
8.2.7 Windows 程序的调试	127
8.2.8 Win32 汇编应用举例	129
8.3 实验内容	135
8.4 实验报告要求	139
附录 A 自选习题与答案	140
附录 B 综合测试题	265
参考文献	270

1.1 实验目的

掌握汇编语言上机过程和 DEBUG 的基本命令及其功能。

1.2 预备知识

1.2.1 汇编语言程序的建立和执行

汇编语言是采用助记符表示相应机器指令的操作码和操作数,并按照一定格式书写的、面向机器的程序设计语言。汇编语言指令与机器指令基本上是一一对应的,其指令只有经过翻译程序(称为汇编程序)翻译后才能变成机器指令。

汇编指令格式: 操作码 操作数

汇编指令举例: `ADD WORD PTR[100],20`

汇编指令与机器指令基本上是一一对应的,但易于理解与记忆,且具有直接和简捷的特点,执行速度快,是唯一能直接控制计算机硬件的语言。

1. 编辑程序

编辑程序是建立源程序(包括各种语言源程序和文本文件)的一种通用工具。源程序的修改也是通过编辑程序进行的。常用的编辑程序有 Edit、Word、记事本等。编程者可以任选一种完成汇编语言源程序的编辑。源程序的文件名一定用 `asm` 扩展名,不可省略也不可更改,否则汇编时出错。另外,如果用 Word,注释用的分号必须是英文分号“`;`”;如果用中文分号“`;`”,汇编时会出错。

2. 汇编程序

汇编程序把汇编语言源文件(`.asm`)转换成二进制代码文件(`.obj`)。

汇编程序有很多种,常用的有小汇编 ASM.EXE、宏汇编 MASM.EXE 和 Turbo 汇编 TASM.EXE 等。小汇编不支持宏操作,因此一般选用 MASM 或 TASM。

假设当前在 C 盘上,源程序文件名为 zk.asm,汇编时输入命令:

```
C:\>masm zk
```

或

```
C:\>masm zk.asm
```

则屏幕显示如下并要求应答。

```
Microsoft(R)Macro Assembler Version 5.00
Copyright(C)Microsoft Corp 1981 - 1985,1987. All rights reserved.
Object filename [ZK.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
```

以上信息中,方括号中信息为该项的默认回答,若不改变默认值,就直接按 Enter 键。

汇编可产生以下三个文件。

① obj 文件。它是汇编后生成的目标代码文件,每一次汇编都应生成这个文件。源程序中如果有错,则不会生成该文件。因此,应特别注意上述信息之后系统给出的 Warning errors(警告错误)和 Severe errors(严重错误)的类型和数目。当严重错误数目不为 0 时,不会生成 .obj 文件,也就无法连接。警告错误数目不为 0 时,可以生成 .obj 文件。若编程者容忍这些警告错误的存在,则可以将此时的 .obj 文件用来连接。

② lst 文件。它是列表文件,可有可无。若不希望生成该文件,则直接按 Enter 键,省略应答。若需要生成列表文件,则在 Source listing 提问行后面给一个文件名即可。lst 文件中同时给出源程序和机器代码程序清单,并给出符号表。这些信息对调试程序有一定的帮助。

③ crf 文件。它是用来产生交叉引用 REF 表的。汇编时生成的 .crf 文件还必须调用另一程序 CREF.EXE,由 CREF.EXE 将 .crf 文件变成 .ref 文件。一般情况下不需要 .crf 文件,因此汇编时在 Cross-reference 提问行后面直接按 Enter 键。

若汇编不产生 .lst 文件和 .crf 文件,可以用下面的命令格式避免屏幕提问,加速汇编。

```
C:\>masm zk;
```

3. 连接程序

经过汇编程序处理而产生的目标文件 .obj 还不能直接运行,因为它还有两个问题,一个是目标文件中还是浮动地址,需再定位;另一个是若为多模块程序,各个模块分别汇编后还需把它们连接起来。因此,连接程序的功能是:

① 找到要连接的所有目标文件。

② 确定所有浮动地址和外部符号所对应的地址。

③ 产生可执行的 .exe 文件。

8086 汇编语言使用的连接程序有 LINK.EXE 和 Turbo 连接程序 TLINK.EXE。对于前面的 ZK.OBJ 其连接命令为:

```
C:\>LINK
```

或

```
C:\>LINK ZK.OBJ
```

则会出现如下的屏幕提示。

```
Microsoft (R) Overlay Linker Version 3.61
Copyright (C) Microsoft Corp 1983 - 1987. All rights reserved.
Run File [ZK.exe]:
List File [NUL.MAP]:
Libraries [LIB]:
```

连接程序可产生以下三个文件。

① exe 文件。它可直接在 DOS 操作系统下运行。若生成与 .obj 文件同名的 .exe 文件,则只要在 Run File 提问行上直接按 Enter 键。

② map 文件。它是连接程序的列表文件,又称为连接映像(LINK MAP)文件。它给出每个段在存储器中的分配情况。一般情况下不需要 .map 文件,所以在 List File 提问行上直接按 Enter 键即可。若需要 .map 文件,则在此行输入文件名再按 Enter 键。

③ lib 文件。它指明程序在运行时所需要的库文件。它不是由连接程序生成的。汇编语言程序无特殊的库文件要求,所以在 Libraries 提问行按 Enter 键即可。但当汇编语言与高级语言接口时,高级语言可能需要一定的库文件,此时输入相应的库文件名即可。

在连接的过程中,也可能出现错误信息。若有错误被检测到,则应回到编辑状态去修改。然后重新汇编和连接,最后生成正确的 .exe 文件(此例为 zk.exe)。

若连接不产生 .map 文件,也不需要 .lib 库文件,则可以用下面的命令格式避免屏幕提问。

```
C:\>link zk;
```

若是多模块,则发出的连接命令格式如下。

```
C:\>link <模块 1> + <模块 2> + ... + <模块 N>
```

各模块应先各自独立汇编生成 .obj 文件,然后由此连接成一个可执行文件。

4. 调试程序

在编写汇编语言源程序时产生的错误,除了一般语法上和格式上的错误可由汇编和连接程序发现指出外,逻辑上的错误都必须用调试程序(DEBUG.COM 或 DEBUG.EXE)来查找。

调试程序 DEBUG 用于试验和检测用户程序。它具备的功能有设置断点和启动地址、单步跟踪、子程序跟踪、条件跟踪、检查修改内存和寄存器、移动内存以及读写磁盘、汇编一行和反汇编等。在 DEBUG 状态下,可对程序进行动态调试,一边运行一边调试,同

时观察各寄存器、内存单元及各标志位的变化情况。

1.2.2 调试程序 DEBUG

DEBUG 是专门为汇编语言设计的一种调试工具,它通过单步、设置断点等方式为汇编语言程序员提供了非常有效的调试手段。

注意: 在 DEBUG 下数值都是十六进制数,后面不用加“H”。

1. DEBUG 程序的调用

在 DOS 提示符下,可输入如下命令:

```
C:\DEBUG [PATH] [文件名] [参数 1] [参数 2]
```

其中:文件名是被调试的文件的名字,它必须是可执行文件(exe 或 com);两个参数是运行被调试文件时所需要的命令参数,在 DEBUG 程序调入运行后,屏幕上会出现提示符“-”,此时可输入所需的 DEBUG 命令。在 DEBUG 控制下,一律使用十六进制数据(无需带 H)。

2. DEBUG 的主要命令

(1) -Q 退出 DEBUG,返回 DOS。

(2) -U 反汇编命令。从当前位置开始反汇编,第 1 次从 CS:IP 位置开始。

-U CS:100 从指定位置开始反汇编。

-U 100 等效于 U CS:100,CS 是默认的段寄存器。

-U CS:100 130 对指定范围反汇编。

-U CS:100 L10 对指定长度反汇编。

(3) -R 检查和修改寄存器内容的命令。

① 显示 CPU 内部所有寄存器内容和标志位状态,格式如下。

-R

R 命令显示中标志位状态的含义如表 1.1 所示。

表 1.1 标志位状态含义

标志名	置位	复位
溢出 Overflow(是/否)	OV	NV
方向 Direction(减量/增量)	DN	UP
中断 Interrupt(允许/屏蔽)	EI	DI
符号 Sign(负/正)	NG	PL
零 Zero(是/否)	ZR	NZ
辅助进位 Auxiliary Carry(是/否)	AC	NA
奇偶 Parity(偶/奇)	PE	PO
进位 Carry(是/否)	CY	NC

② 显示和修改某个寄存器的内容,格式如下。

- R 寄存器名

例如输入: -R AX

系统将响应如下:

AX 0000

:

表示 AX 当前的内容是 0000,此时若不对其修改,可按 Enter 键,否则,输入要修改的内容。如:

- R BX

BX 0211

:043E

则 BX 的内容修改为 043E。

③ 显示和修改标志位的状态,格式如下。

- RF

(4) -D 显示内存单元的内容的命令。显示当前段当前位置开始的内存区(内存区默认长度为 128 个字节)中存储单元的内容,第一次使用时,默认为 CS 段。

-D DS:10 显示从指定位置上开始内存区的存储单元内容。

-D DS:10 50 显示指定范围的内存区的存储单元内容。

-D DS:10 L20 显示指定长度的内存区的存储单元内容。

-D 100 显示当前段(上一次使用的段寄存器)内偏移量 100H 处的内存区的内存单元内容。

-D CS:100 显示指定段的内存区。

(5) -E 修改内存单元内容,默认为 DS 段。

-E DS:10 显示/修改 DS:10 处一字节内容。

-E 10 显示/修改 DS:10 处一字节内容,按空格键相继修改下一字节的内容,按 Enter 键结束。

(6) -G 从当前 CS:IP 的位置执行程序。

-G=CS:100 从指定位置执行程序。

-G=100 等效于 G=CS:100,CS 是默认值。

-G 200 从当前位置执行,设 200 为断点。

-G=0 10 20 从 CS:0 起执行,10,20 是两个断点。

(7) -T 从当前 CS:IP 处执行一条指令。

-T 3 从当前 CS:IP 起执行 3 条指令。

-T=CS:10 从 CS:10 起执行 1 条指令。

-T=10 等同于 T=CS:10,CS 是默认值。

(8) -P, -P 3, -P=CS:10, -P=10 类似于 T 命令,但是把子程序当做一条指令,在子程序执行完毕后停下。

(9) -A 汇编命令,从指定地址开始输入汇编语句,把它们汇编成机器代码相继存放在从指定地址开始的存储器中。

-A 从当前 CS:IP 处开始汇编。

-A 10, -A CS:10 从 CS:10 处开始汇编。

(10) -I 读指定端口的内容。

-I 21 读出并显示 21H 号端口的内容。

(11) -O 向指定的端口写数据。

-O 21 00 向 21H 号端口写入 00H。

(12) -? 显示 DEBUG 命令的格式。

3. 进入 DEBUG 环境

单击“开始”→“运行”(也可以直接按 **Ctrl**+**r** 打开“运行”窗口)。在“运行”窗口中输入命令。此时输入 DEBUG,就进入 DEBUG 程序了。界面上显示一个连字符“-”。利用本小节介绍的 DEBUG 的各种命令,程序员就可以编辑、执行、检查和修改程序了,以下是一个简单的过程演示。

(1) 准备输入汇编指令,使用 DEBUG 命令 a,并按 Enter 键,出现:

```
- a
- 0B75:0100_ (光标闪烁)
```

注意:因机器不同会使前面的地址不一样。

假设输入下面的指令:

```
0B75:0100 MOV AX,1
0B75:0103 MOV BX,2
0B75:0106 ADD AX,BX
0B75:0108
```

最后一行(即 0108)什么也不输入,直接按 Enter 键就退出 a 命令。

(2) 执行汇编指令。输入 t 3(3 表示 3 条命令,有 5 条就输入 5,不输入默认为 1),然后按 Enter 键就可以执行前面输入的 3 条汇编指令了。

```
- t 3
AX = 0001 BX = 0000 CX = 0000 DX = 0000 SP = FFEE BP = 0000 SI = 0000 DI = 0000
DS = 0B75 ES = 0B75 SS = 0B75 CS = 0B75 IP = 0103 NV UP EI PL NZ NA PO NC
0B75:0103 BB0200 MOV BX,0002
AX = 0001 BX = 0002 CX = 0000 DX = 0000 SP = FFEE BP = 0000 SI = 0000 DI = 0000
DS = 0B75 ES = 0B75 SS = 0B75 CS = 0B75 IP = 0106 NV UP EI PL NZ NA PO NC
0B75:0106 01D8 ADD AX,BX
AX = 0003 BX = 0002 CX = 0000 DX = 0000 SP = FFEE BP = 0000 SI = 0000 DI = 0000
DS = 0B75 ES = 0B75 SS = 0B75 CS = 0B75 IP = 0108 NV UP EI PL NZ NA PE NC
0B75:0108 06 PUSH ES
```

执行过程和结果如图 1.1 所示。

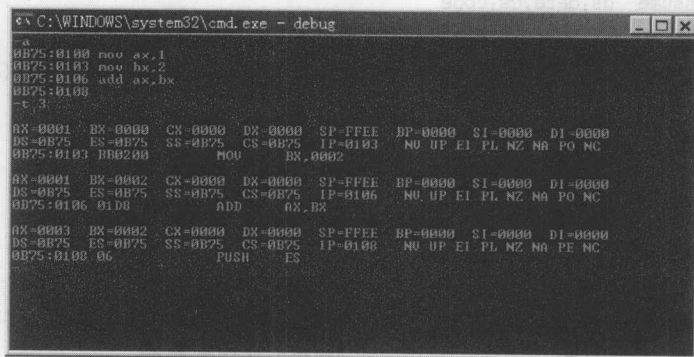


图 1.1 调试程序 DEBUG 的演示

例 1.1 求平均数程序的上机过程。汇编程序要求：在数据段中定义一个 5 字节数据的数组 array，把它们看做有符号数并求它们的平均数，结果保存在 avg 内存单元。

分析：求平均数的程序框图如图 1.2 所示。

上机过程如下。

(1) 调用文本编辑程序 Edit 建立 asm 文件

Edit 是 DOS 环境下的编辑工具，当然读者也可以使用其他的 W32 编辑工具，如 EditPlus、Ultra edit、NotePad 等软件在 Windows 下建立源文件。下面假设启动的是 DOS 系统或启动 Windows 切换到 Command 命令窗口(为便于讨论下面均以 DOS 说明)，汇编语言上机的工作环境在 E:\masm 子目录下。输入以下命令：

```
E:\masm\edit sample.asm
```

使用该命令建立以 sample.asm 为文件名的源文件如下所示。然后保存文件，返回 DOS 系统。

```
; *****
data        segment                ; 定义数据段
    array    db    1,2,3,4,5
    avg       db    ?
data        ends

; *****
code        segment                ; 定义代码段
; -----
```

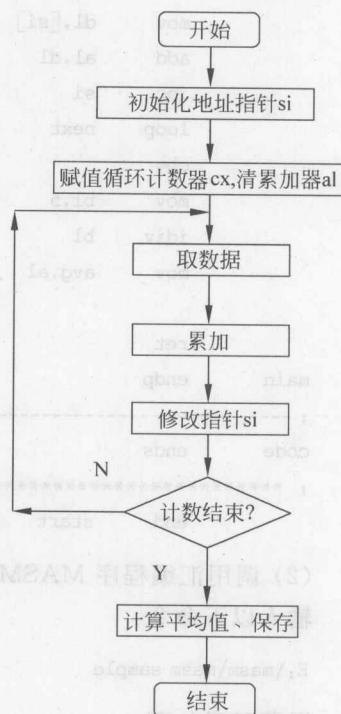


图 1.2 例 1.1 的程序框图