



普通高等教育“十二五”规划教材



计算机硬件技术 实践指导

程启明 黄云峰 杨艳华 编

书中实验参考程序均可免费下载。

下载网址: <http://jc.cepp.sgcc.com.cn>



中国电力出版社
CHINA ELECTRIC POWER PRESS

013026162



普通高等教育“十二五”规划

计算机硬件技术 实践指导

程启明 黄云峰 杨艳华 编
王保义 主审



中国电力出版社
CHINA ELECTRIC POWER PRESS

TP303-43
125



北航

C1633428

内 容 简 介

本书是普通高等教育“十二五”规划教材。

计算机硬件技术是一门实践性很强的课程,学习时必须理论联系实际,亲自动手做实验才能达到预期的目的。本书是配合《计算机硬件技术》而编写的实践指导教材。以 8086/8088 CPU 为背景,整合了微机原理、接口技术和微机系统的软/硬件实验及课程设计。全书有汇编语言程序设计的软件实验、基于硬件平台的微机及接口电路的硬件实验、基于 Proteus 仿真软件的微机及接口电路的硬件实验、微机原理的课程设计共四部分 8 章内容。为了方便学生实验与学习,本书还有 8 个附录,且所有实验的源程序代码都有电子课件。

本书内容新颖,可作为各类高等学校(包括本科、高职高专)非计算机专业的计算机硬件技术基础、微机原理及应用、微机接口技术等课程的实践教材,从事计算机应用开发的科技人员也可以参考。

图书在版编目(CIP)数据

计算机硬件技术实践指导 / 程启明, 黄云峰, 杨艳华编. —北京: 中国电力出版社, 2012.12

普通高等教育“十二五”规划教材

ISBN 978-7-5123-3863-0

I. ①计… II. ①程… ②黄… ③杨… III. ①硬件—高等学校—教学参考资料 IV. ①TP303

中国版本图书馆 CIP 数据核字 (2012) 第 304076 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

*

2013 年 1 月第一版 2013 年 1 月北京第一次印刷

787 毫米×1092 毫米 16 开本 13.75 印张 332 千字

定价 24.50 元

敬告读者

本书封底贴有防伪标签,刮开涂层可查询真伪

本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究

前言

本书是《计算机硬件技术》一书的配套实践教材。“计算机硬件技术”或“微机原理及应用”是很多高校非计算机专业重要的基础课程之一。该课程是一门实践性强的课程，其中很多的原理、规则、现象等仅仅靠学习教科书是无法完全掌握的，必须通过大量的实践才能比较直观和深刻地理解。在进行课程实践过程中，可以让学生体验分析问题、提出解决方案、通过编程等手段实现解决方案、不断调试，最终达到设计要求的全过程，从而帮助学生系统地掌握微机原理接口技术的相关知识，达到将知识融会贯通的目的。

本实践指导书编写的目的就是为了提高学生的实践能力，提高汇编等语言的编程能力及对接口等硬件的理解、分析能力和设计接口电路的能力，从而能学以致用。学生只有通过实际编程和微机及接口的硬件实践，才能真正掌握软/硬件设计的方法，从中获益和提高。

本书以 8086/8088 CPU 为背景，整合了微机原理、接口技术和微机系统的软/硬件实验或课程设计项目。全书共四个部分 8 章，主要内容包括：第 1、2 章为汇编语言程序设计的软件实验部分；第 3、4 章为基于硬件平台的微机及接口电路的硬件实验部分；第 5、6 章为基于 Proteus 仿真软件的微机及接口电路的硬件实验部分；第 7、8 章为微机原理的课程设计部分。其中，每个实验一般都包含实验目的、实验类型、实验内容及步骤、硬件连线、软件参考流程、实验结果分析及思考题等。另外，本书还有附录部分，附录中包括实验要求、8086/8088 汇编指令、汇编程序出错信息、DEBUG 命令、ASCII 码表、DOS 系统功能调用、BIOS 中断调用、Proteus VSM 下 8086 的元件库等内容。此外，为了方便学生学习，本书的所有实验源程序代码都有电子课件（在中国电力出版社网站上下载）。

本书在内容安排上注重系统性、循序性、逻辑性、科学性、实用性和先进性。本书最大的特点是新颖性和完整性，也就是把当今最先进的微机两个仿真软件 Proteus 和 Emu8086 引入教学实践中，突出最新的硬件模拟技术和实用性；书中既有传统的软件实验和硬件平台实验内容，又有 Proteus 仿真实验和课程设计内容。

本书与其他《计算机硬件技术》（或《微机原理及应用》）实践教材有以下 3 点主要区别：①采用传统硬件实验平台与 Proteus 仿真软件平台两套并行的方式来做微机硬件及接口实验，将 Labcenter 公司开发的 Proteus（7.5 以上版本）仿真软件引入到微机的硬件教学中，从而使学生通过 Proteus 软件可灵活搭建、自由组合各种复杂的微机系统，仿真过程“所见即所得”，实验操作环境直观；②汇编语言实验部分将 Emu8086 模拟器引入到 8086 的软件实验中，从而使学生的汇编软件实验更容易编程与调试；③把本课程的实验内容与课程设计内容分为前后两个部分，承上启下地连在一起学习，从而加深学生对微机硬件的实践能力。

本书内容丰富、概念清晰、实用性和可操作性强，是学习微机原理、汇编与接口技术课程的一本较好的实践教材。本书建议的学时数为 10~30。

本书可作为各类高等学校（包括本科、大中专、高职班）非计算机专业的计算机硬件技术基础、微机原理及应用、微机接口技术等课程的实践教材，也可以作为从事计算机应用开发的科技人员的参考用书。

本书由程启明、黄云峰、杨艳华 3 位老师负责编写，其中程启明老师负责编写第四部分的课程设计及附录，并全面负责本书的统稿工作；黄云峰老师负责编写第三部分基于 Proteus 仿真软件的微机硬件实验部分内容；杨艳华老师负责编写第一部分汇编语言的软件实验部分和第二部分基于硬件平台的微机硬件实验部分。华北电力大学计算机系王保义教授审阅了本书，并提出了许多宝贵的意见和建议，在此对他表示深深地谢意。在本书的编写过程中，借鉴了一些教材的编写经验和网上公开资料，在此谨向这些作者表示诚挚地感谢！

限于编者水平，书中难免有不妥之处，敬请广大读者批评指正，以便再版时及时修正。如果本教材或网站上实验源程序在使用过程中出现问题，请发邮件至：chengqiming@shiep.edu.cn。

编 者
2013 年 2 月

目 录

前言

第一部分 汇编语言程序设计软件实验部分

第 1 章 汇编语言程序设计软件系统平台	1
1.1 EDIT/MASM/LINK/DEBUG 4 个小软件的作用	1
1.2 Masm for Windows 集成实验环境的使用	4
1.3 Emu8086 汇编软件的使用	8
1.4 Visual Studio 的 IDE 汇编软件的作用	15
第 2 章 8086/8088 汇编语言软件实验	21
2.1 数据运算类实验	21
2.2 查找和排序类实验	24
2.3 键盘和窗口类实验	27
2.4 数码转换类实验	30

第二部分 8088 硬件接口实验部分

第 3 章 8088 硬件实验系统平台	33
3.1 8088 技术指标要求	33
3.2 8088 系统资源分配	34
3.3 通用外围电路	35
第 4 章 8088 硬件接口实验	37
4.1 存储器读/写实验	37
4.2 8255A 可编程并行口实验	38
4.3 8253A 定时/计数器实验	39
4.4 8251A 的串行接口应用实验	40
4.5 8259A 的单级中断控制实验	41
4.6 采用 ADC0809 的 A/D 转换实验	43
4.7 采用 D/A 转换器 DAC0832 产生方波实验	45
4.8 采用 D/A 转换器 DAC0832 产生锯齿波实验	46
4.9 8279A 可编程键盘显示接口实验	46
4.10 小直流电动机调速实验	49
4.11 步进电动机控制实验	50
4.12 继电器控制实验	51

4.13	8237A 可编程 DMA 控制器实验	52
4.14	8259A 串级中断控制实验	54
4.15	USB 接口 CH375 应用实验	56
4.16	采用 8253 和 8255 实现电子表实验	57
4.17	采用 D/A、A/D 实现闭环测试实验	57
4.18	采用 0832 和 8255 实现对直流电动机的调速控制实验	59
4.19	中断次数计数器实验	60
4.20	16C550 串行口控制器实验	60
4.21	液晶显示控制实验	62
4.22	点阵 LED 显示实验	64

第三部分 微机及接口电路的硬件仿真实验部分

第 5 章	Proteus 仿真软件系统平台	69
5.1	Proteus 基本使用方法	69
5.2	Proteus ISIS 下 8086 的仿真	80
第 6 章	基于 Proteus 的 8086 接口实验	89
6.1	基本 I/O 应用—I/O 译码	89
6.2	波形发生器——8253 的应用	92
6.3	键盘与数码管——8255A 的应用（数字量输入/输出）	95
6.4	中断应用——8259A 的应用	99
6.5	模数转换——ADC0808 的应用	102
6.6	数模转换——DAC0832 的应用	106
6.7	串行通信——8251A 的应用	108
6.8	液晶显示的控制——HD44780 的应用	111
6.9	LED16×16 点阵显示——74HC373 的应用	114
6.10	直流电动机控制实验——8255A 的应用（PWM 脉宽调制）	117
6.11	步进电动机控制实验——8255A 的应用（环形脉冲控制）	120

第四部分 课程设计部分

第 7 章	课程设计的要求	124
7.1	课程设计的目的和意义	124
7.2	课程设计的指导及要求	124
7.3	课程设计的设计过程	125
7.4	课程设计的组织形式及设计步骤	126
7.5	课程设计的时间进度安排	126
7.6	课程设计的答辩	127
7.7	课程设计的考核方法及成绩评定标准	127

7.8	课程设计的报告内容及格式要求	128
第 8 章	课程设计的课题及举例	131
8.1	课程设计的出题原则	131
8.2	课程设计的参考题目	131
8.3	课程设计举例	146
附录 A	实验要求与实验报告格式规范	185
A.1	实验预习	185
A.2	实验操作	185
A.3	实验总结及完成实验报告	185
A.4	实验注意事项	185
附录 B	8086/8088 汇编指令速查手册	187
B.1	数据传送类指令	187
B.2	算术运算指令	187
B.3	逻辑运算指令	188
B.4	串指令	188
B.5	程序转移指令	189
B.6	处理器控制指令	190
B.7	伪指令	190
附录 C	汇编程序出错信息表	191
附录 D	DEBUG 启动及基本命令	194
D.1	DEBUG 程序的启动	194
D.2	DEBUG 的主要命令	194
附录 E	ASCII 码表	199
附录 F	DOS 系统功能调用	201
附录 G	BIOS 中断调用	206
附录 H	Proteus VSM 仿真的元件库及常用元件说明	208
	参考文献	211



第一部分 汇编语言程序设计软件实验部分

第1章 汇编语言程序设计软件系统平台

汇编语言的开发软件有4种类型,它们分别是:①采用 EDIT/MASM/LINK/DEBUG 4 个小软件;②采用 Masm for Windows 集成实验环境;③采用 Emu8086 汇编软件;④采用 Visual Studio 的 IDE 软件。下面就来介绍这4种汇编语言软件的上机过程。

1.1 EDIT/MASM/LINK/DEBUG 4 个小软件的作用

该方法需要安装4个小软件程序,它们分别是文本编辑程序(如 DOS 的 EDIT.COM、Windows 的记事本软件等的纯文本格式的编辑器)、汇编程序(如宏汇编 MASM.EXE、小汇编 ASM.EXE、TASM.EXE 等)、连接程序(如 LINK.EXE、TLINK.EXE 等)和调试程序(如 DEBUG.EXE、CV.EXE 等)。这些小程序需要在 DOS 状态下运行。先将上述4个小程序放在 C:\ASM 文件夹下面,打开“开始”→“程序”→“附件”→“命令提示符”进入 DOS 状态。图 1-1 为采用命令提示符操作方法的过程示意图。

当汇编语言源程序编好后,要实现其功能,需经过建立、汇编、链接与运行、调试4个阶段过程。

1. 采用编辑程序,建立汇编源程序.asm 文件

源程序就是用汇编语言编写的程序,它不能被机器识别。源程序必须以.asm 为文件扩展名。通过 EDIT.EXE 文本编辑器进行输入,运行 EDIT,其操作界面便会出现于屏幕上(见图 1-2),可在提示符下输入源程序,当输入完毕后,选择存盘并给输入的文件起一个文件名,格式为 filename.asm,其中 filename 为起的文件名,由1~8个字符组成;.asm 是为汇编程序识别而必须加上去的,不可更改。当然,也可用其他文本编辑器进行录入、编辑,最后将文件存为 filename.asm 的形式即可。

2. 采用汇编程序,汇编成目标文件.obj 文件

汇编语言源程序经过汇编,才可以生成目标程序,这个过程由汇编程序实现。其基本功能是把用汇编语言书写的源程序翻译成机器语言的目标代码、检查用户源程序中的错误且显示出错信息、生成列表文件等。汇编程序 MASM 的格式为

```
MASM filename
```

其中 filename 为第1步中建立的文件名。这时汇编程序的输出文件有目标文件名(.obj),列表文件名(.lst),交叉引用文件名(.crf)3个,便会出现3次提问,在这一路按 Enter 键即可。汇编过程结束时,会给出程序中的警告性错误 Warning Errors 和严重错误 Servers Errors,前者指一般性错误,后者指语法性错误,当存在这两类错误时,屏幕上除指出错误个数外,

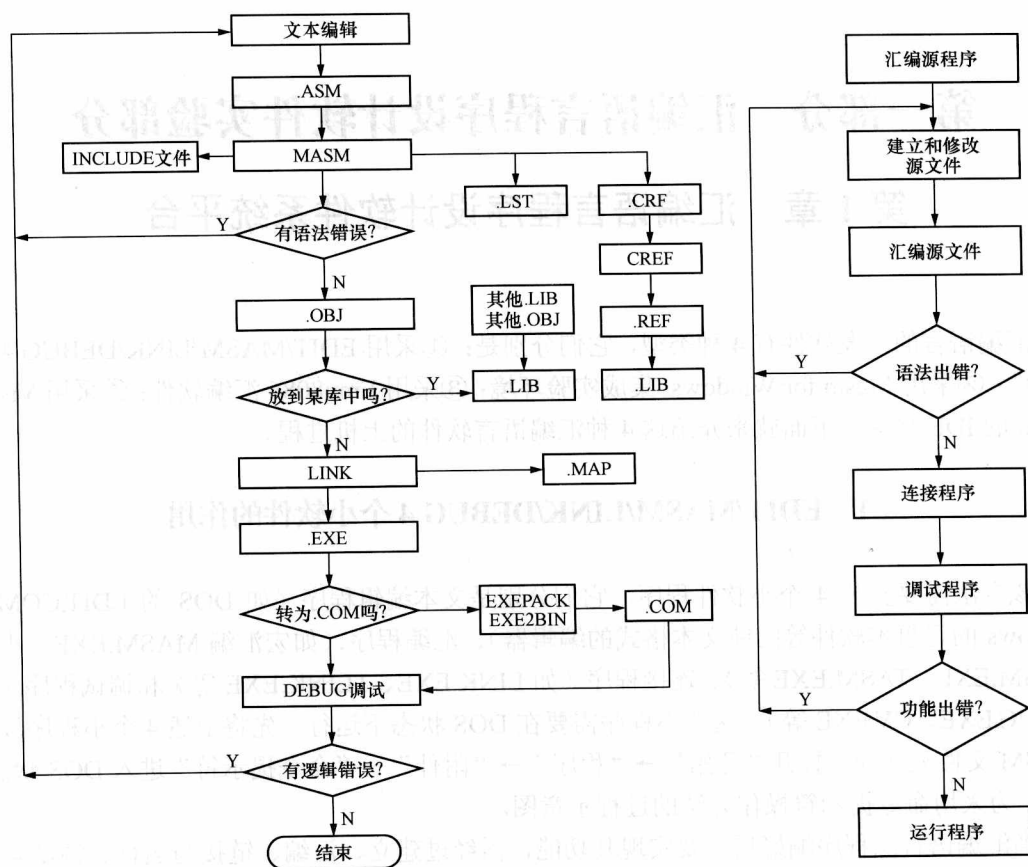


图 1-1 采用命令提示符操作方法的过程示意图

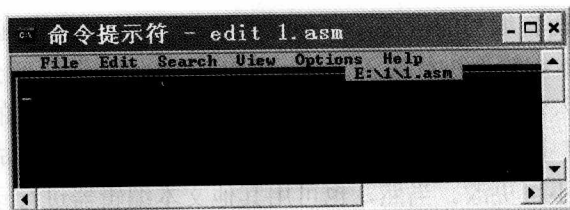


图 1-2 EDIT 编辑界面

还给出错误信息代号（附录 B 为汇编程序出错信息表），程序员可以通过查找手册弄清错误的性质。如果汇编过程中，发现有错误，则程序员应该重新用编辑命令修改错误，再进行汇编，直到汇编正确通过。图 1-3 为 MASM 编译无错误时的界面。注意，汇编过程只能指出程序中的语法错误，并不能指出算法错误和其他错误。

3. 采用连接程序，得到可执行文件（.exe 文件）

汇编过程根据源程序产生出二进制的目标文件（.OBJ 文件），但 .OBJ 文件用的是浮动地址，不能直接上机执行，还必须使用连接程序（LINK.EXE）将 OBJ 文件转换成可执行的 EXE 文件。LINK 命令还可以将某一个目标文件和其他多个模块（这些模块可以由用户编写的，也可以是某个程序中存在的）链接起来。连接程序 LINK 命令的格式为

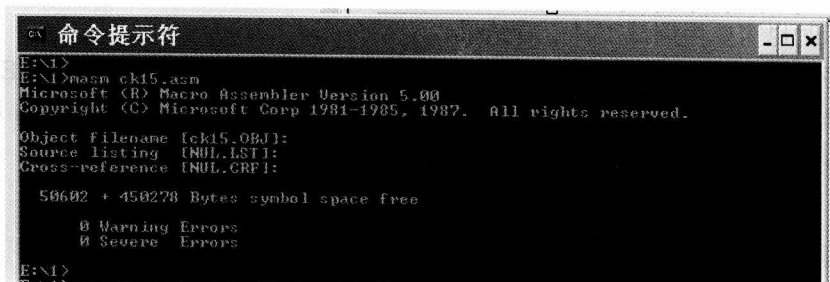


图 1-3 MASM 编译文件无错误时的界面

LINK filename

Filename 为.OBJ 输入文件，用户程序有时会用到库函数，此时，对于提示信息 Libraries [L.IB]，要输入库名。如果连接没有错误，则 LINK 过程产生.exe 可执行文件、.map 的列表分配文件（也称映像文件）两个输出文件，映像文件给出每个段在内存中的分配情况。图 1-4 为 LINK 连接 1.OBJ 文件界面。

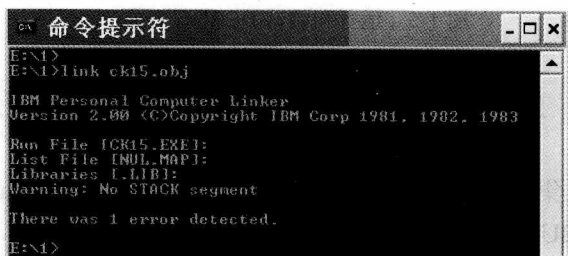


图 1-4 LINK 连接 1.OBJ 文件界面

从 LINK 过程的提示信息中，可看到最后给出了一个“无堆栈段”的警告性错误，这并不影响程序的执行。当然，如果源程序中设计了堆栈段，则无此程序。

如果.OBJ 文件有错误，连接时会指出错误的原因。对于无堆栈警告（warning: no stack segment）的提示，可以不予理睬，它是由于在源程序中没有定义堆栈段的原因，对于比较小的源程序和不需要再特别定义堆栈段的源程序，可以不定义堆栈段，它并不影响程序的正确执行。但如果连接时有其他的错误，则要检查并修改源程序.asm，然后再重新汇编 MASM.exe，连接 LINK.exe 的步骤，直到得到正确的.exe 文件为止。

当生成.exe 文件后，就可以输入该文件名运行了，注意不必输入扩展名。看它是否按所设想的那样得出结果。在试运行期间，要尽量试一些临界状态，看程序是否运行稳定、结果是否正确。试运行程序时，有可能产生一些莫名其妙的结果，说明程序有逻辑错误，还需进入下一步继续调试。实际上，大部分程序必须经过调试才能纠正程序设计中的逻辑错误，从而得到正确的结果。

从 6.0 版以后，Microsoft 公司把 MASM 和 LINK 的功能由一个 ML.EXE 程序完成，只需一个命令就可把源程序汇编连接生成.EXE 文件。

4. 采用调试程序，重新修改程序

静态查错即检查源程序，并在源程序级用文本编辑器进行修改，然后再汇编、连接、运行。但有时静态检查不容易发现问题，这时就需要使用调试工具动态查错。当程序结果不能

在屏幕上显示时也需要用调试工具查看结果。

调试程序 DEBUGEXE 来进行程序调试、检查错误。图 1-5 为运行和调试 DEBUG 界面。发现错误并修改程序后, 还需经过编辑、汇编、链接来纠正错误。关于 DEBUG 程序中的各种命令, 可参见附录 C。



图 1-5 运行和调试 DEBUG 界面

1.2 Masm for Windows 集成实验环境的使用

Masm for Windows 集成开发环境 (Integrated Development Environment, IDE) 相当于把 EDIT/MASM/LINK/DEBUG 4 个小软件集成为一个软件, 它使用简单、方便, 很适合汇编语言的初学者使用。

1. 打开 Masm for Windows 的 IDE

依次单击“开始/程序/汇编语言集成实验环境 Masm for Windows IDE”菜单进入“Masm for Windows 集成实验环境”。

2. 输入汇编程序

在图 1-6 的“程序输入区”中输入一个“1+2”结果等于 3 的汇编程序, 软件本身有一个汇编程序框架。

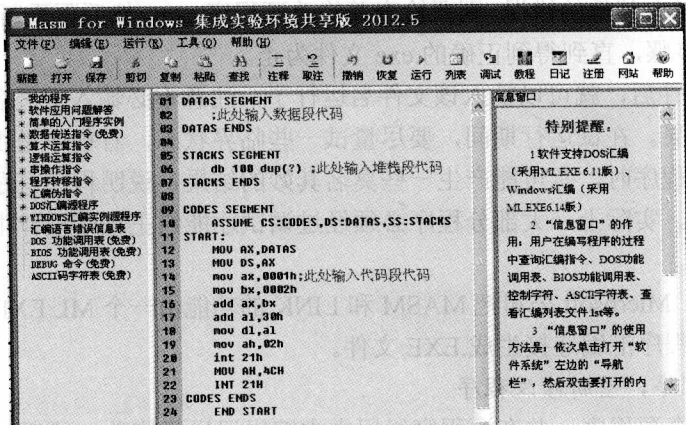


图 1-6 Masm for Windows 的集成实验环境

3. 保存汇编程序

单击“工具栏”中的“保存”按钮（或“文件/另存为”或“文件/保存”），弹出“另存为”对话框，输入文件名，如“第一个汇编程序”，然后单击“保存”按钮即可。

4. 运行程序

单击“工具栏”中的“编译生成目标文件”按钮，可以编译并生成链接文件。单击“生成可执行文件”完成链接。最后单击“运行”按钮，即可出现程序的运行结果，如图 1-7 所示。

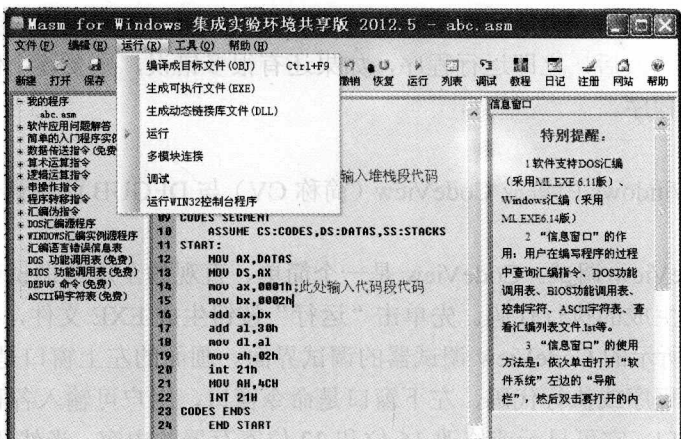


图 1-7 程序编译、链接、运行

程序运行结果的下面 Press any key to continue 表示按任意键退出 DOS 窗口，如图 1-8 所示。

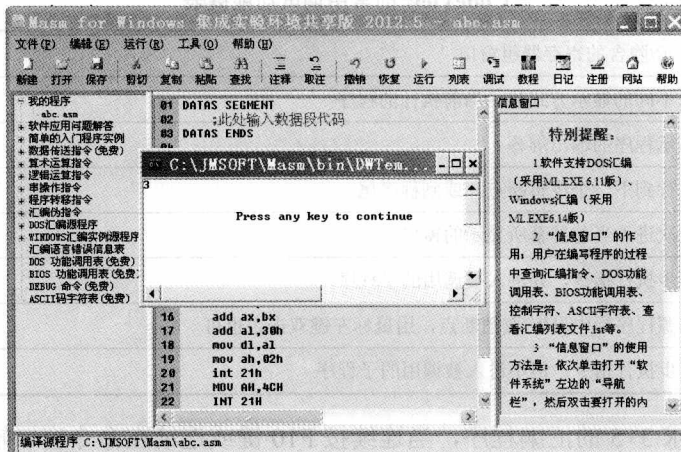


图 1-8 程序运行结果

5. 修改程序语法错误

当运行程序发生语法错误时，Masm for Windows 集成实验环境会自动定位源程序中错误所在行的位置并高亮显示该行，修改好第一条发生错误行后，双击任一条错误信息，该软件定位源程序中与之相对应错误所在行的位置并高亮显示该行，以便改正错误。表 1-1 为汇编程序的语法错误信息形式。

表 1-1 汇编程序的语法错误信息形式

文件名	行号	冒号	错误编号	冒号	错误内容
↓	↓	↓	↓	↓	↓
C:\汇编程序文件夹\第一个程序.asm	(9)	:	Error A2008	:	Syntax error:CODES

一条语句错误可能会产生若干条错误信息，例如图 1-7 中就是因为第 9 行 `SEGMENT` 多输入一个 `S`，导致很多错误，只要将第 9 行 `SEGMENTS` 改为 `SEGMENT` 即可调试通过。一般情况下，第一条错误信息最能反映错误的位置和类型，所以调试程序时务必根据第一条错误信息进行修改，修改后，立即运行程序，如果还有很多错误，则要逐个修改，即每修改一处错误要运行一次程序。

6. 调试程序

在 Masm for Windows 中集成 CodeView（简称 CV）与 DEBUB 两种调试工具，默认为用 CV 调试程序。

（1）采用 CodeView 调试。CodeView 是一个简单、直观的全屏幕调试工具，它可调试多种语言的源程序所生成的执行代码。先单击“运行”按钮生成 EXE 文件，再单击“调试”按钮，会出现图 1-9 所示的 CodeView 调试器的调试界面。画面的左上窗口是调试器的主窗口，其显示被调试的源程序或执行代码，左下窗口是命令窗口，用户可输入各种 `DEBUG` 命令，右窗口是寄存器窗口，它可显示或修改 16 位和 32 位寄存器的内容。当然还有其他窗口，如：内存窗口、查看内容窗口（`Watch`）和程序输出窗口（`View`）等。表 1-2 为 CodeView 的常用调试功能键表。

表 1-2 CodeView 的常用调试功能键表

F2	显示/隐含的寄存器组窗口
F3	以不同的显示方式显示当前执行的程序
F4	显示程序的输出屏幕
F5/F7	执行到下一个逻辑断点，或到程序尾
F6	依次进入当前屏幕所显示的窗口
F8	单步执行指令，并进入被调用的子程序
F9	在源程序行中设置/取消断点，用鼠标左键双击之也可
F10	单步执行指令，但不进入被调用的子程序

图 1-9 中调试求 `3+5` 的汇编程序，当连续按 F10 键或连续在命令窗口输入 `P` 命令执行到 `ADD AL, 03` 时，可以看到 `AL` 的值为 8。

（2）用 `DEBUG` 调试。

1）依次单击“工具/选项”，在出现图形界面上，选中“`DEBUG` 调试”，再单击“确定”按钮。

2）单击“运行”按钮生成 EXE 文件，再单击“调试”按钮，出现图 1-10 所示的界面。图 1-10 是调试求 `3+5` 的汇编程序，当连续输入 `P` 命令执行到 `ADD AX, BX` 时，可以看到 `AL` 的值为 3。

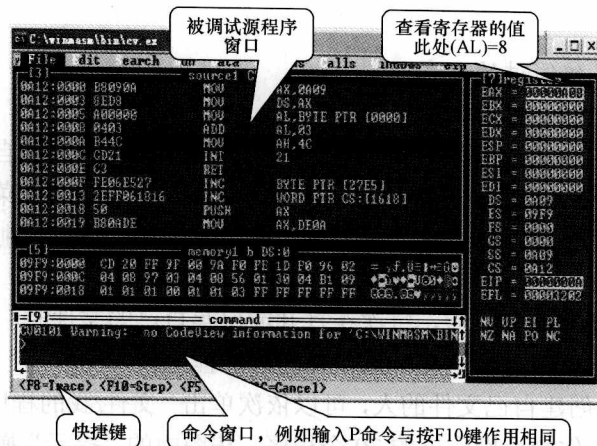


图 1-9 CodeView 调试器的调试界面

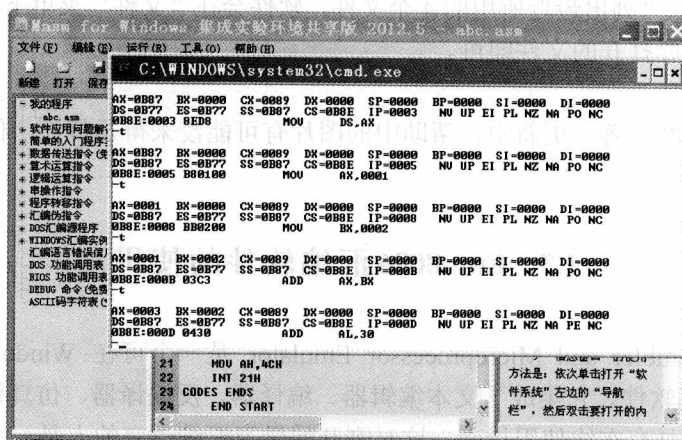


图 1-10 DEBUG 调试界面图

7. 在线帮助

当编写程序时, 有可能忘记某个指令的用法, 只要用鼠标单击关键字, 按鼠标右键在弹出的菜单上选实时帮助或按 F2 键即可获得该指令的帮助。

8. 编辑程序

(1) 在编辑程序时, 建议用右键菜单实现“剪切、复制、粘贴、查找、替换、撤销与恢复”等功能。

(2) 利用撤销与恢复功能更正错误。

9. 快速找到程序中的某一行语句

方法 1: 当程序行数比较少时, 可以通过鼠标滚轮或用鼠标拖动程序输入区的垂直滚动条的滑块, 程序的行号也会随之滚动, 即可找到程序中要找的行。

方法 2: 单击鼠标的右键, 在弹出的菜单中选择“定位到行”, 在打开的对话框中输入行号, 再单击“确定”按钮即可找到该行。

10. 编写良好风格、可读性强的程序

利用软件的智能缩进功能, 无须人为地添加或删除空格, 就可以编写出良好风格、可读

性强的程序。

11. 快速打开自己编写过的程序

可采用下面 3 种方法, 推荐采用第 1 种方法。

方法 1 (推荐使用): 在编写自己的程序之前, 先创建好一个存放自己的程序的文件夹, 然后在“Masm for Windows 集成实验环境”中依次单击“工具/选项”菜单, 打开“选项”对话框, 再单击“设置”按钮, 在弹出的“浏览文件夹”对话框中找到刚创建好的文件夹, 进行设置即可。

设置好“我的程序文件夹”后, 每次在打开或保存自己的文件时, 软件会自动定位到自己设置好的文件夹, 可以很方便地打开或保存自己的程序。

方法 2: 对于没有创建自己文件的人, 可以依次单击“文件/我的程序”, 打开“我的程序”对话框, 在这里保存着你最近操作过的 30 个程序, 按图中的“提示”操作, 即可打开要找的程序。

方法 3: 在勾选“列出最近所用的 4 个文件, 软件会在“文件”菜单下列出最近所用的 4 个文件, 从中选择要打开的文件即可。

12. 提示

该软件正在不断完善、更新中, 帮助中的图片有可能没来得及替换, 但不会影响读者按帮助操作此软件。

1.3 Emu8086 汇编软件的使用

Emu8086-Assembler and Microprocessor Emulator 是一个可在 Windows 环境下运行的 8086CPU 汇编仿真软件。它集成了文本编辑器、编译器、反编译器、仿真调试、虚拟设备和驱动器为一体, 并具有在线使用指南, 这对刚开始学习汇编语言的人是一个很有用的工具。可以在仿真器中单步或连续执行程序, 其可视化的工作环境让用户操作更容易。可以在程序执行中动态观察各寄存器、标记位以及存储器中的变化情况。仿真器会在模拟的计算机中执行程序, 以避免程序运行时到实际的硬盘或内存中存取数据。此外, 该软件完全兼容 Intel 新一代处理器, 包括 Pentium III、Pentium IV 的指令。

1. 软件启动

双击桌面上的 Emu8086 的图标, 出现启动界面如图 1-11 所示, 用户可以选择新建文本、程序实例、快速指南、近期文档。

软件提供的程序实例中包含了几十种典型的程序代码, 其中包括数值计算、逻辑运算、屏幕显示、键盘输入、文件打印、电动机控制、温度控制、交通信号灯控制等。

在软件提供的快速指南中提供了多种在线帮助工具, 其中包括文件索引、8086 CPU 指令使用指南、系统中断调用列表以及用法等。用户可以通过该工具快速地掌握 8086 CPU 指令体系和 Emu 8086 汇编真软件的使用。

2. 新建文件

单击图 1-11 中的 New 选项, 软件会弹出如图 1-12 所示的选择界面。

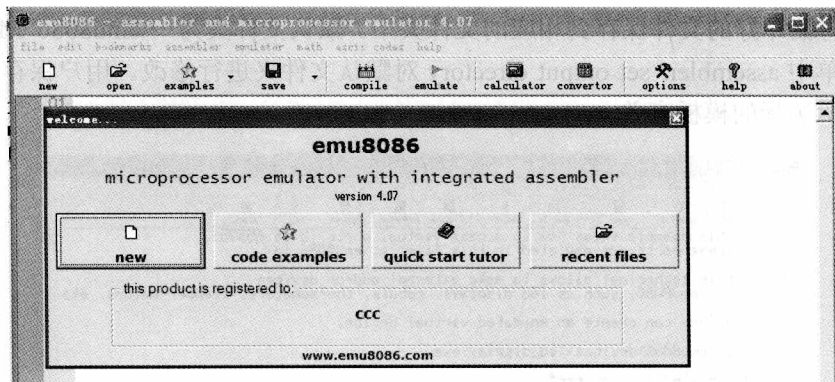


图 1-11 启动界面

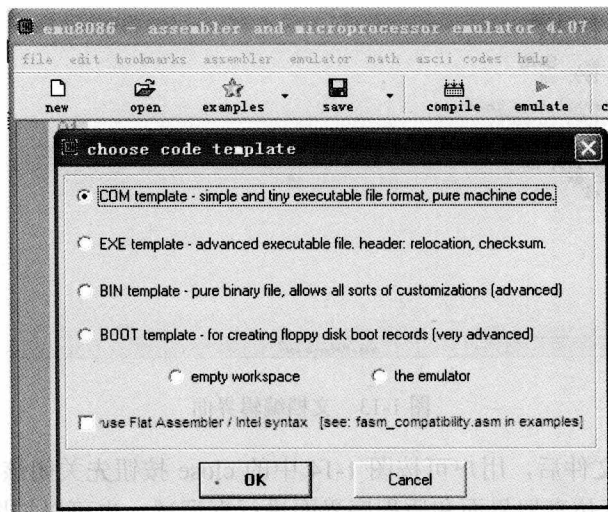


图 1-12 新建文本格式选择

(1) COM 模板。适用于简单且不需分段的程序，所有内容均放在代码段中，程序代码默认从 ORG 0100H 开始。

(2) EXE 模板。适用于需分段的复杂程序，内容按代码段、数据段、堆栈段划分。需要注意的是采用该模板时，用户不可将代码段人为地设置为 ORG 0100H，而应由编译器自动完成空间分配。

(3) BIN 模板。二进制文件，适用于所有用户定义结构类型。

(4) BOOT 模板。适用于在软盘中创建文件。

此外，若用户希望打开一个完全空的文档，可选择 empty workspace 单选项。

3. 编译和加载程序

用户可在上述选择的模板中编写程序，如图 1-13 所示。该编辑界面集文档编辑、指令编译、程序加载、系统工具、在线帮助为一体，其菜单功能见表 1-3。

编写完程序后，用户只需单击工具栏上的 compile 按钮，即可完成程序的编译工作，并弹出如图 1-14 所示的编译状态界面。若有错误，会在窗口中提示，若无错误还会弹出保存界