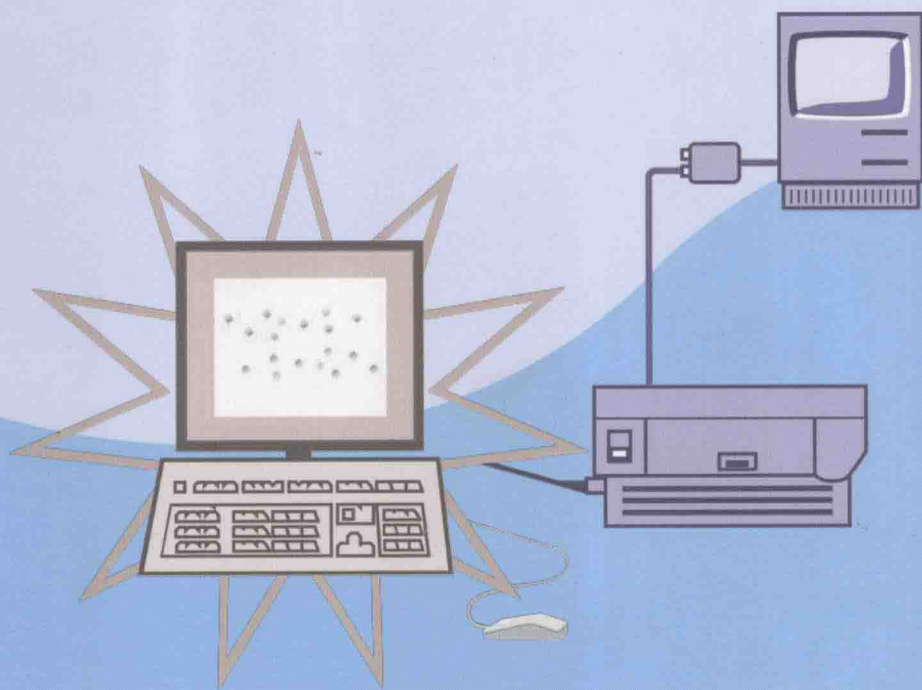


汇编语言程序设计 实验教程

主 编 付 玉

副主编 王智文 黎 珍

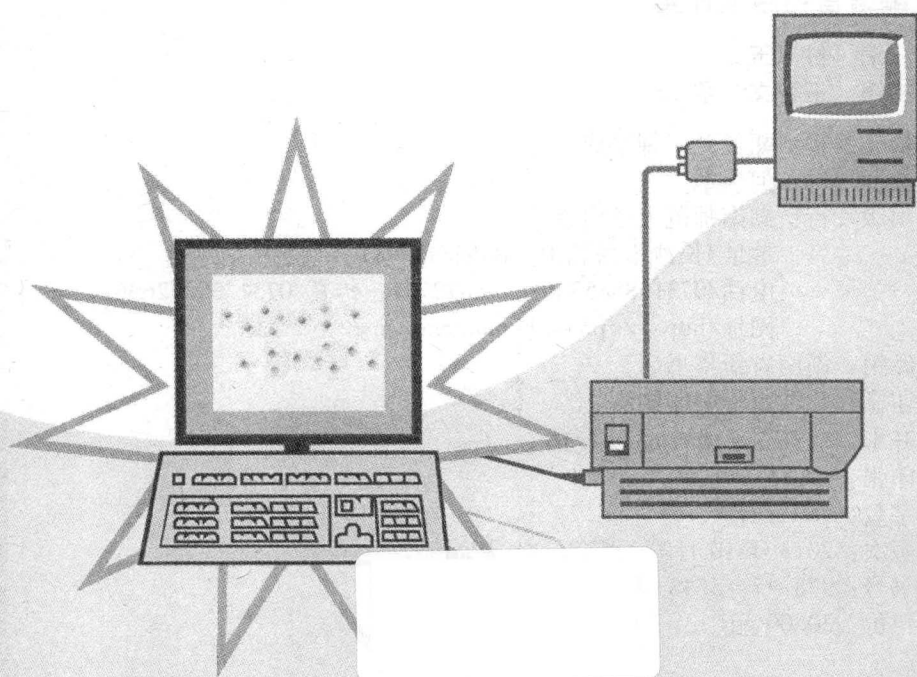


湖南师范大学出版社

汇编语言程序设计 实验教程

主 编 付 玉

副主编 王智文 黎 珍



湖南师范大学出版社

图书在版编目 (CIP) 数据

汇编语言程序设计实验教程/付玉主编. —长沙: 湖南师范大学出版社, 2013. 10

ISBN 978 - 7 - 5648 - 1400 - 7

I. ①汇… II. ①付… III. ①汇编语言—程序设计—高等学校—教材 IV. ①TP313

中国版本图书馆 CIP 数据核字 (2013) 第 242458 号

汇编语言程序设计实验教程

主 编 付 玉

副主编 王智文 黎 珍

◇责任编辑: 胡亚兰 钟坚成

◇责任校对: 何远翠

◇出版发行: 湖南师范大学出版社

地址/长沙市岳麓山 邮编/410081

电话/0731. 88853867 88872751 传真/0731. 88872636

网址/http: //press. hunnu. edu. cn

◇经销: 湖南省新华书店

◇印刷: 长沙市华中印刷厂

◇开本: 787mm × 1092mm 1/16

◇印张: 19. 75

◇字数: 460 千字

◇版次: 2013 年 10 月第 1 版第 1 次印刷

◇书号: 978 - 7 - 5648 - 1400 - 7

◇定价: 30. 00 元

前 言

汇编语言程序设计是大学生计算机基础课程，也是一门实践性很强的课程。汇编语言上机实验部分一直是这门课程的重点和难点。汇编程序设计如何编写、如何运行、如何调试等，学生仅靠理论上的钻研是不可能完全掌握好的，还必须通过上机实验来检测和验证。要具备实验条件并不难，只要有计算机就能够进行。然而如何在具体学习实践中熟练掌握程序的编写、改错、上机调试等实践操作，如何一步步深入编程、调试，进一步过渡到单模块调试和多模块调试，如何由基本程序设计过渡到综合程序设计等，需要一个系统化、流程化的指导才可以进行。根据现代教育的建构主义教学观，只有创设良好的学习情景，才可以激发学生主动探索、主动运用知识的积极性，才能进行知识的构建，从而实现有意义地、高效地学习。本书正是基于这一理论而编写的。

本书将书本指导和计算机辅助教学紧密结合起来，有助于帮助学生理解汇编语言的基本原理，锻炼学生的动手能力，提高学生综合应用知识的能力。依据循序渐进、由浅入深的规律，通过实例复习巩固所学知识，并开展汇编语言程序的进一步设计。通过这种既具有构造性又具有创造性的学习活动，更有利于初学者理解和掌握汇编语言程序设计的过程及窍门，通过上机实验，并在实验后完成实验报告，又能进一步加深和拓宽所学知识，引导读者在掌握基本的汇编语言程序设计的基础上，进而能自主地进行综合性设计实验。

本书主要介绍了 DOS 环境和 Windows 环境下汇编语言程序设计的实验步骤，以及 Debug 常用命令及用法；汇编语言程序设计中常见的几种程序设计方法；算术运算类操作实验方法；字符串操作及输入/输出实验方法；输入输出程序设计；中断程序设计；综合程序设计实验中指导学生进行排序和检索程序设计；系统调用程序设计；模块化程序设计；语言接口程序设计；图形处理程序设计；声音程序设计；通信接口程序设计；磁盘文件管理程序设计来加强学生的汇编语言程序设计的综合开发能力。书中给出了实验程序清单，便于学生学习和对比。书中还指导学生将汇编语言程序设计与 C/C++ 语言进行混合编程，进一步提升学生的程序开发能力。此外，本书也对汇编语言程序上机操作的常见错误及其修正进行了详细说明，有助于学生减少编程过程中产生的错误。本书选用了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。

全书共分 6 章。其中，第 1 章通过一个典型实例引导读者熟悉汇编语言程序上机操作的全过程，并进一步介绍有关实用程序的使用和操作。第 2 章和第 3 章分别包括 8 个基础性实验和 8 个综合性设计实验，实验内容丰富，伸缩性大，便于教学过程中教师进行选择。第 4 章提供了所有实验程序的清单，供读者编程参考。第 5 章设计了汇编语言程序设计与 C/C++ 言进行混合编程，将汇编语言程序设计的执行高效率性和 C/C++ 程序对硬件

的处理效率有机地结合起来，有利于进一步提升学生的编程能力。第6章分类列出了常见上机出错信息、出错原因分析及修正方法。

本书是针对全国绝大部分高校工科院校计算机类专业课程《汇编语言程序设计》及《微机原理与汇编语言程序设计》的教学大纲和普遍采用的汇编语言程序设计教材而编写的，可用于大学本、专科生的“计算机原理”、“汇编语言程序设计”等相关课程的实验、实训通用性教材，也可独立使用，亦可作为有关技术人员的参考书籍。

全书由湖南师范大学工程与设计学院的付玉博士、副教授担任主编，由广西科技大学计算机学院王智文博士、教授担任副主编。其中，付玉老师编写了全书的第2章、第4章和第6章；王智文老师编写了第1章、第3章和第5章。付玉老师的硕士研究生黎珍同学负责了全书的代码调试及录入工作，以及第6章部分代码的设计工作，并对全书的文本进行了第一次校核。全书在编写过程中得到了湖南师范大学出版社的大力支持，亦得到作者所在单位，即湖南师范大学工程与设计学院和广西科技大学计算机学院的大力协助，同时参考了相关书籍或文献，在此一并致谢！书中如有错误或不当之处，欢迎读者批评指正。

作 者

2013年9月

目 录

第 1 章 汇编语言程序设计上机操作介绍	(1)
1.1 熟悉实验环境	(1)
1.2 上机操作：了解实验环境	(2)
1.2.1 进入命令提示符窗口	(2)
1.2.2 进入 C:\MASM 文件夹	(2)
1.3 汇编语言程序的上机过程介绍	(3)
1.4 上机实例简介	(5)
1.4.1 了解上机程序	(5)
1.4.2 利用记事本输入并编辑汇编语言源程序	(5)
1.4.3 用 MASM 程序编译源文件，产生目标文件	(7)
1.4.4 用 LINK 程序连接目标文件，产生可执行程序	(9)
1.4.5 用 DEBUG 程序调试与运行可执行程序	(10)
1.5 MASM 汇编程序的使用	(13)
1.5.1 宏汇编 MASM 的使用	(13)
1.5.2 列表文件和交叉索引文件	(15)
1.6 LINK 连接程序的使用	(16)
1.6.1 会话方式	(16)
1.6.2 命令方式	(19)
1.6.3 映像文件与子程序库	(19)
1.7 DEBUG 程序的使用	(21)
1.7.1 可执行文件的运行	(21)
1.7.2 DEBUG 调试程序的启动	(22)
1.7.3 DEBUG 命令详解	(23)
1.7.4 GDB 中断点设置	(43)
1.7.5 GDB 中数据查看	(44)
1.7.6 GDB 中单步执行	(45)
1.8 上机自测	(46)
1.8.1 实验程序 1	(46)
1.8.2 实验程序 2	(46)

第2章 汇编语言程序设计基础实验	(47)
实验 2.1 顺序结构程序设计	(47)
实验 2.2 分支结构程序设计	(49)
实验 2.3 循环结构程序设计	(52)
实验 2.4 子程序设计	(54)
实验 2.5 算术运算程序设计	(56)
实验 2.6 字符处理程序设计	(57)
实验 2.7 输入输出程序设计	(59)
实验 2.8 中断程序设计	(61)
第3章 综合性设计性实验	(63)
实验 3.1 排序和检索程序设计	(63)
实验 3.2 系统调用程序设计	(65)
实验 3.3 模块化程序设计	(66)
实验 3.4 语言接口程序设计	(67)
实验 3.5 图形处理程序设计	(71)
实验 3.6 声音程序设计	(72)
实验 3.7 通信接口程序设计	(74)
实验 3.8 磁盘文件管理程序设计	(79)
第4章 实验参考程序	(82)
4.1 上机自测实习(1.8节)实验参考程序	(82)
4.2 基础实验参考程序	(83)
4.2.1 实验 2.1 顺序结构程序设计实验参考程序	(83)
4.2.2 实验 2.2 分支结构程序设计实验参考程序	(88)
4.2.3 实验 2.3 循环结构程序设计实验参考程序	(91)
4.2.4 实验 2.4 子程序设计实验参考程序	(97)
4.2.5 实验 2.5 算术运算程序实验参考程序	(108)
4.2.6 实验 2.6 字符处理程序实验参考程序清单	(111)
4.2.7 实验 2.7 输入输出程序实验参考程序清单	(114)
4.2.8 实验 2.8 中断程序实验参考程序清单	(117)
4.3 综合性设计性实验参考程序清单	(121)
4.3.1 实验 3.1 排序和检索程序设计参考程序清单	(121)
4.3.2 实验 3.2 系统调用程序设计参考程序清单	(124)
4.3.3 实验 3.3 模块化程序设计实验参考程序清单	(126)
4.3.4 实验 3.4 语言接口程序设计参考程序清单	(135)
4.3.5 实验 3.5 图形处理程序设计参考程序清单	(138)
4.3.6 实验 3.6 声音程序设计参考程序清单	(141)

4.3.7 实验 3.7 通信接口程序设计参考程序清单	(144)
4.3.8 实验 3.8 磁盘文件管理程序设计参考程序清单	(145)
第 5 章 汇编语言与 C/C++ 混合编程实验	(153)
5.1 汇编语言与 C/C++ 混合编程简介	(153)
5.2 汇编语言与 C/C++ 混合编程相关规则	(157)
5.3 汇编语言与 C/C++ 混合编程实验实例	(163)
第 6 章 汇编语言程序上机操作常见错误及其修正	(295)
6.1 汇编操作常见错误及修正	(295)
6.2 LINK 操作常见错误及修正	(302)
6.3 DEBUG 操作常见错误及修正	(304)
参考文献	(308)

第1章 汇编语言程序设计上机操作介绍

要进行一门程序设计语言的实验，首先应了解这个语言的实验环境和上机过程，本章通过一个简单的例子，帮助读者快速熟悉上机步骤。最后让读者自己上机做一做，进一步巩固上机操作的过程。

1.1 熟悉实验环境

计算机语言实验环境就是使用该计算机语言所需要的硬件环境和软件环境。具体来说，就是使用者用到的机型、操作系统，所需要的语言系统软件，以及确定具体编程时源程序、目标程序、可执行程序所存放的路径和文件夹。这是实验前必须首先具备或解决的问题。

对汇编语言程序设计来说，目前绝大多数应用是基于 MASM 宏汇编语言。它需要什么样的实验环境呢？

先看看用到的汇编语言系统软件：

汇编语言编译器：它的作用是将汇编语言源程序（.ASM 文件）翻译为目标代码程序（.OBJ 文件）。具有这个功能的文件有：MASM.EXE、ML.EXE、CV.EXE 等，本书使用的是较为普遍的 MASM 5.0 及以上版本。

连接器：它的作用是连接目标代码程序和库函数代码生成可执行程序文件（.EXE 文件），这个软件的文件名是 LINK.EXE。

可执行程序动态调试器：它的作用是对可执行程序进行装载情况的静态了解和动态执行调试，具有这个功能的文件有：DEBUG.EXE、TR.EXE 等，本书使用的是较为普遍的 DEBUG 程序。

上述软件需要在 PC 系列微型计算机上的 DOS 操作系统下运行。今天用到的 PC 系列微型计算机都具备该运行环境。在 Windows 操作系统上有命令提示符窗口，该窗口提供了模拟的 DOS 操作系统环境。

接下来要安排有关文件的路径和文件夹。

一般语言系统文件集中在一个文件夹中，例如本书把它放在 C:\MASM 中。

具体编程时源程序、目标程序、可执行程序又放在何处呢？你可以把它们集中存放在一个文件夹中，例如 C:\MYFILE 中，这时，所有涉及这些文件的路径前缀就不能省略了。比如你要对存放在 C:\MYFILE 中的 EX1.ASM 程序进行编译，如果当前目录是 C:\

MASM, 编译命令可能是:

MASM C: \ MYFILE \ EX1. ASM

或者, 如果当前目录是 C: \ MYFILE, 编译命令可能是:

C: \ MASM \ MASM EX1. ASM

这对初学者可能是个麻烦, 因为它涉及文件全名中的路径省略问题。如果你对 DOS 操作非常熟悉, 也许带来的仅仅是多输入一些字符而已。为了使读者能集中精力于编程实验学习中, 我们将语言系统文件和用户上机生成的源程序、目标程序、可执行程序等文件统统放在一个文件夹里。如无特别声明, 这个文件夹就是 C: \ MASM, 并把它作为默认路径。

1.2 上机操作: 了解实验环境

1.2.1 进入命令提示符窗口

在 Windows 平台上, 有三种方式可进入命令提示符窗口。

第 1 种方式: 按照“开始→程序→附件→命令提示符”顺序打开命令提示符窗口。

第 2 种方式: 按照“开始→运行”顺序打开“运行”对话框(图 1-1)。在对话框中输入 cmd, 点击“确定”, 即进入命令提示符窗口。

第 3 种方式: 直接双击 DOS 环境下开发的可执行程序(.exe 或 .com 程序), 此时系统进入命令提示符窗口并执行相关程序。

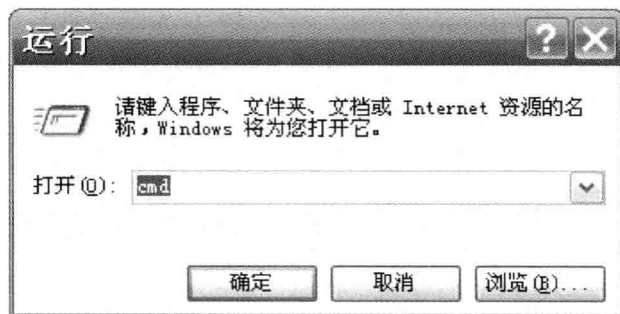


图 1-1 Windows 系统的“运行”对话框

小窍门: 建议用第 2 种方式, 因为第 1 次输入 cmd 后, 以后会保留原来的输入, 这样再次进入就只需点击“确定”了, 比第 1 种方式要简便些。

1.2.2 进入 C: \ MASM 文件夹

在命令提示符窗口中键入 cd \ masm 并回车, 该命令要求将当前目录改变为 masm。

我们可以查看在这个文件夹下是否具备所需要的文件。为此, 键入命令 dir 并回车, 该命令要求将当前目录中的所有文件列出来, 见图 1-2。

图中列出了15个文件，即：汇编程序 MASM.EXE、连接程序 LINK.EXE 和调试程序 DEBUG.EXE。这3个文件是必不可缺的，没有它们，汇编语言实验就无法进行下去。

此外还有：索引程序 CREF.EXE、库管理程序 LIB.EXE 以及1个汇编语言源程序 test.asm。

随着用户上机操作的进行，将在该文件夹中不断产生新的文件。

读者也许有一个疑问，输入和编辑汇编语言源程序的程序在哪儿呢？一般来说，计算机语言源程序是 ASCII 码文本文件，它可以应用 Windows 系统中的记事本程序，也可以使用 DOS 系统中的 EDIT 程序来编辑。EDIT 程序由于常驻内存，因此不会在文件列表中“显身”。但在命令提示符窗口下，不管在哪一个文件夹下都可以启动该程序。

```

C:\WINDOWS\system32\cmd.exe
驱动器 C 中的卷是 WINXP
卷的序列号是 2A1D-0905

C:\masm 的目录

2012-10-27 08:31 <DIR>      .
2012-10-27 08:31 <DIR>      ..
1987-07-31 08:00          15,830 CREF.EXE
2000-01-10 20:00          20,634 debug.exe
1994-09-16 15:00        134,144 LIB.EXE
1987-07-31 08:00          39,100 LINK.EXE
1987-07-31 08:00          65,557 MASM.EXE
2012-10-27 07:58           1,254 test.asm
2012-10-27 08:32           149 TEST.OBJ
2012-10-27 08:33           553 TEST.EXE
2012-10-30 07:30           1,631 1.ASM
2012-10-30 07:42           197 1.OBJ
2012-10-30 07:43          4,730 1.EXE
2012-10-30 07:43           224 1.LST
2012-11-28 22:46           721 2.asm
2012-11-28 22:47           141 2.OBJ
2012-11-28 22:47           561 2.EXE
          15 个文件          285,426 字节
           2 个目录 2,397,782,016 可用字节
  
```

图 1-2 应用 dir 命令列出当前目录中的所有文件

1.3 汇编语言程序的上机过程介绍

了解了汇编语言程序的上机环境，接下来应该熟悉汇编语言程序的上机过程。

下面以 MASM 汇编软件为例来进行说明，假定该软件位于 C:\MASM 文件夹下，所有操作都以该目录为当前目录。

汇编语言源程序可能只是一个源程序文件，也可能是由多个源程序文件构成。我们先介绍单程序文件的上机操作，然后再讲复杂程序的模块化构建。

整个上机过程包括建立与编辑源程序、源程序汇编产生目标文件、连接目标程序产生可执行程序、调试运行可执行程序四个步骤。上机操作流程如图 1-3 所示。

步骤 A：建立与编辑源程序

采用 Windows 系统中的“记事本”或命令提示符状态下的 EDIT 文本编辑器建立并编辑汇编语言源程序（扩展名为 .asm）。

步骤 B：汇编（即编译）

在命令提示符状态下，采用 MASM 程序对源程序汇编（即编译），产生目标程序（扩展名为 .obj）。

如果源程序有语法错误，则编译会通不过，必须重新修改源程序，直到通过汇编产生目标程序为止。

步骤 C：连接

在命令提示符状态下，采用 LINK 程序对目标程序进行连接，生成可执行程序（扩展名为 .exe）。

如果连接通不过，必须重新审视编辑、编译、连接过程，改错直到通过连接产生可执行程序为止。

步骤 D：运行并调试

由于汇编语言指令系统涉及输入输出安排较复杂，一般程序运算结果以及中间结果往往存放于内存或寄存器中。因此，初学汇编语言程序设计的用户，要了解程序运算结果就必须直接深入内存调试程序。方法是，在“命令提示符”状态下，采用 DEBUG 程序对可执行程序进行运行调试，如果程序运行结果达不到要求或出现异常（即出现逻辑错误），则可能要修改源程序，重新进行编辑、编译、连接和运行调试。

我们将通过一个简单的上机实例说明一个汇编语言程序的输入、编辑、编译、连接、运行与调试的全过程。在 1.5 节中读者可以选择学习有关 EDIT、MASM、LINK 和 DEBUG 的相关操作，并在上机实验时独立完成一个或多个程序的上机操作全过程。

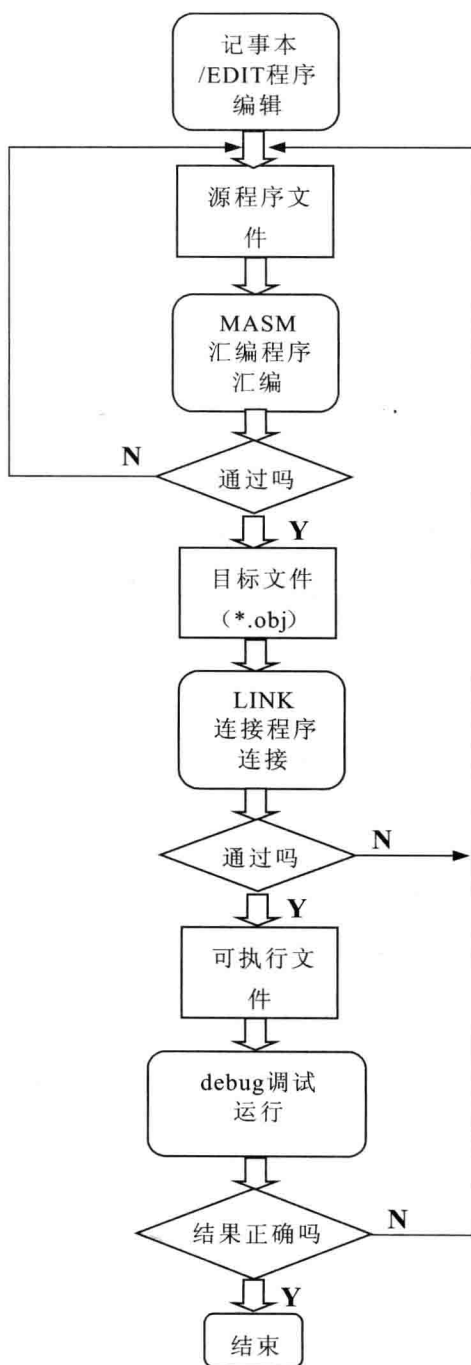


图 1-3 汇编语言程序上机操作流程

1.4 上机实例简介

下面举例说明一个汇编语言程序的输入、编辑、编译、连接、运行与调试的全过程。为了简化过程描述，假定在用户目录 C: \ MASM 下存放有 MASM 宏汇编程序、LINK 连接程序和 DEBUG 调试程序。

1.4.1 了解上机程序

我们编写一个简单的程序，它的功能是实现 $W = X + Y$ 。下面是程序清单，为了帮助读者理解程序，对有关程序行作了注释：

```
DATA    SEGMENT ; 建立一个名为
                ; DATA 的数据段
X    DB    0E5H ; 定义变量 X
Y    DB    34H  ; 定义变量 Y
W    DW    ?    ; 定义变量 W
DATA    ENDS    ; DATA 数据段结束
CODE    SEGMENT ; 建立一个名为 CODE 的代码段
        ASSUME CS: CODE, DS: DATA ; 说明与有关的段寄存器相联系的段
START:  MOV AX, DATA ; 装填 DS 段地址
        MOV DS, AX
        XOR AH, AH    ; AH 清零，为高位字节加作准备
        MOV AL, X
        ADD AL, Y      ; AX = X + Y
        ADC AH, 0      ; 进位存入高位字节
        MOV W, AX      ; 相加结果存放于变量 W 中
        MOV AH, 4CH    ; 返回 DOS
        INT 21H
CODE    ENDS    ; CODE 代码段结束
        END START    ; 程序结束
```

1.4.2 利用记事本输入并编辑汇编语言源程序

记事本是 Windows 系统提供的一个用于编写文本文件的程序。文本文件是一种典型的顺序文件，该文件的内容纯粹是由 ASCII 码字符或中文字符组成。文本文件中除了存储文件有效字符信息（包括能用 ASCII 码字符表示的回车、换行等信息）外，不能存储其他任何信息，包括声音、动画、图像、视频等信息。各种计算机语言源程序都是文本信息，都适于用记事本来编辑。由于记事本具有 Windows 操作风格，因此有些计算机操作基础的人很容易掌握；而且它能编辑中文，在程序中插入中文注释就很方便。建议读者优先选用记

记事本输入并编辑汇编语言源程序。

图 1-4 显示了记事本操作的界面。从上至下依次为标题栏、菜单栏、文本编辑窗口、状态栏。菜单栏中的“文件”、“编辑”功能操作同一般的 word 文档操作。状态栏右侧能显示当前光标所在的位置（用行序号 Ln，列序号 Col 来表示），很方便程序阅读和查错。

记事本操作步骤如下：

① 按照“开始→程序→附件→记事本”顺序打开“记事本”窗口。

小窍门：可将记事本图标拖到 Windows 桌面上，便于快捷启动。

② 在编辑窗口输入程序如图 1-4，注意每输入一程序行，就一定要按回车键。一个程序行可分成标号列、指令助记符列、操作数列、注释列四部分，最好每一列左对齐，这样格式清楚，便于阅读。

注意：汇编语言程序中的字符和数据不区分大小写，除非是字符或字符串数据。

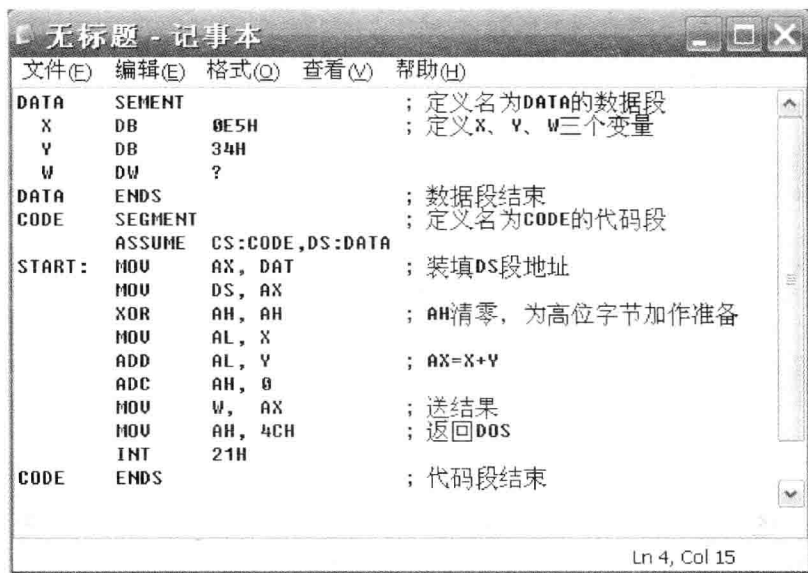


图 1-4 在记事本编辑窗口输入程序

小窍门：利用 TAB 键，很容易做到程序行的各列左对齐。例如，输入程序行：

START: MOV AX, DATA ; 装填 DS 段地址

操作步骤：光标位于行首，键入“START:”，输入标号；按 TAB 键，光标移至该行的第 2 区，键入指令助记符“MOV”；接着按 TAB 键，光标移至该行的第 3 区，再键入“AX, DATA”；接下来输入注释行，按 TAB 键，光标移至该行的第 4 区，即注释列，键入分号，将语言栏改为中文状态，再键入“装填 DS 段地址”并按回车键即结束该行的输入。其他各行输入类似，就能实现各列左对齐了。注意，这里的分号是注释行标记，只能在英文状态下输入，且不能是中文字体！

小窍门：你可以充分利用 Windows 风格的编辑操作，实现字符串的复制、删除和移动等操作，加快输入和编辑速度。

③ 给输入的程序命一个文件名，其扩展名必须为 .ASM。假定文件名为 test1.asm。

操作过程是：用鼠标选择“文件→保存”打开“保存”对话框。在上方“保存在”

一栏中选择 MASM 文件夹，将下方“保存类型”选为“所有文件”，在“文件名”一栏输入文件名 test1.asm（见图 1-5），最后按“保存”按钮，就将当前输入的程序命名为 test1.asm，并存放在 MASM 文件夹中。

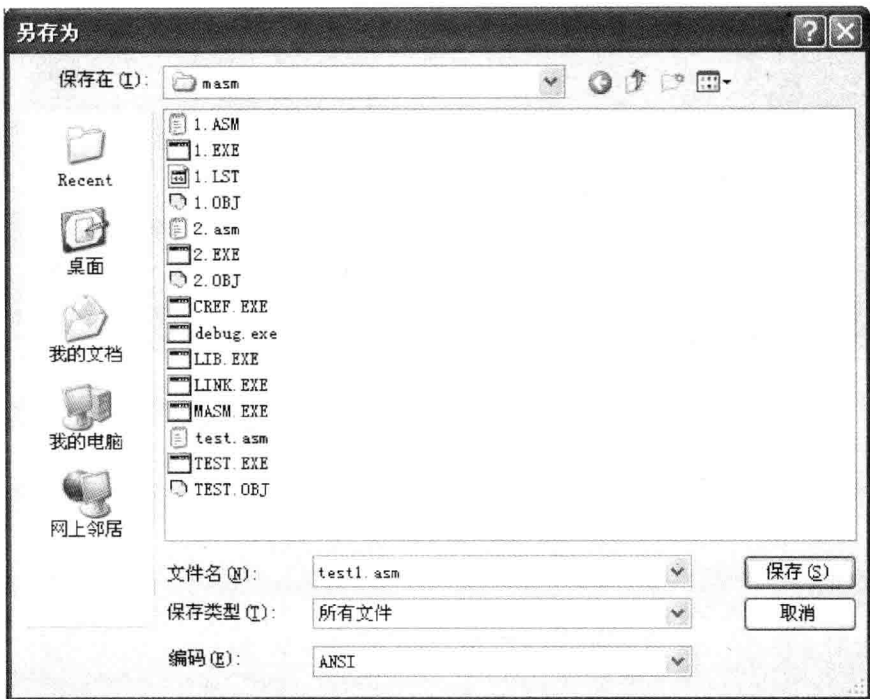


图 1-5 保存文件对话框

注意：

① 一定要将保存类型选择为“所有文件”，如果不选，则系统按默认的 .TXT 文件保存，而非 .ASM 文件了。

② 为了能显示状态栏，应不选择记事本“格式”菜单下的“自动换行”功能，这样就可以在“查看”菜单下选定“状态栏”，从而让记事本窗口下方出现状态栏。

1.4.3 用 MASM 程序编译源文件，产生目标文件

我们采用一种简便的方法操作此过程。

进入命令提示符窗口，并进入 C:\MASM。键入“masm test1;”并回车（注意“;”号不能省！由于编译的源程序是 .ASM 文件，文件扩展名可以省略）。

此时屏幕显示见图 1-6。

表明编译没有通过，因为虽然没有警告错误（0 Warning Errors），但是存在 7 个致命的编译错误（7 Severe Errors）。只要有一个致命错误，编译就通不过，也不会产生目标程序代码文件。

这 7 个致命的编译错误是：

程序第 1 行：错误码 A2105：可能是指令或目录错；

程序第 2、3、4 行：错误码 A2086：所表达的数据没有段表示；

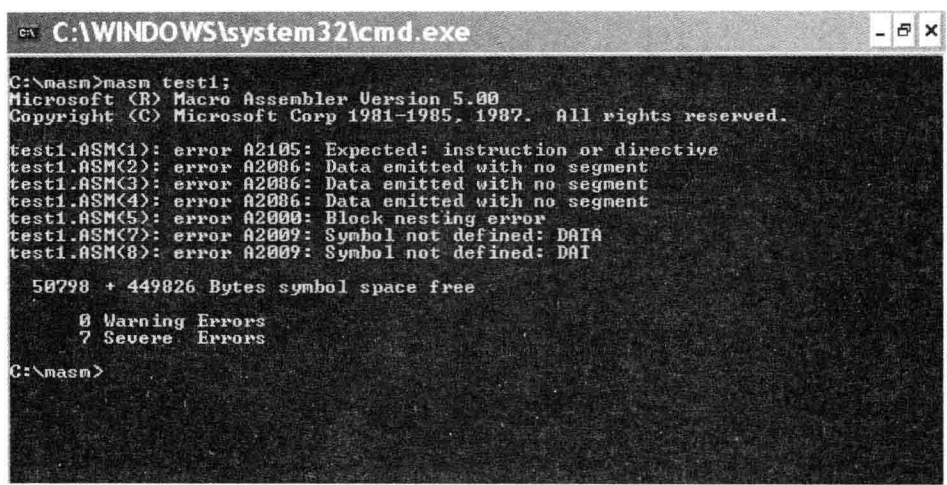


图 1-6 MASM 编译后的界面

程序第 5 行：错误码 A2000：模块嵌套错误；

程序第 7 行：错误码 A2009：符号 DATA 未定义；

程序第 8 行：错误码 A2009：符号 DAT 未定义；

难道程序中真有这 7 行错误？我们再次进行编辑操作，对原程序进行修改。为此进入记事本窗口，打开 test1.asm 文件。

仔细审视源程序 1~5 行和 7~8 行，发现第 1 行应为“SEGMENT”，第 8 行末尾的“DAT”应为“DATA”，由于段（SEGMENT）表示不正确，使得系统对于与数据段名“DATA”相关的第 2、3、4 行产生了出错疑问。

将光标移到第 1 行，插入字符“G”。

将光标移到第 8 行末尾，键入“A”。

选“文件→退出”，出现图 1-7 所示的对话框：

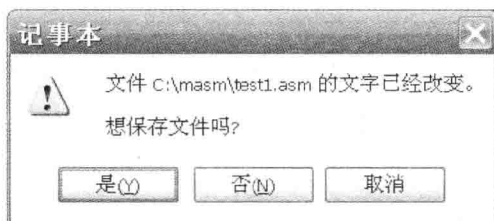


图 1-7 退出记事本时要求文件存盘的对话框

选“是”，则修改后的源程序文件存盘并退出记事本。

然后重新用“masm test1;”命令对源程序进行编译。

此时系统显示有 0 个致命的编译错误（0 Severe Errors），0 个警告错误（0 Warning Errors）。表明编译已通过，并产生目标文件 TEST1.OBJ。

小窍门：如果既能显示记事本编辑界面，同时又显示图 1-6 的界面，二者对照就容易找到出错处进行程序修改了。这个不难，我们可以事先保存源程序文件而不退出记事本窗口，再适当调整记事本窗口和命令提示符窗口，就能实现二者对照（见图 1-8）。

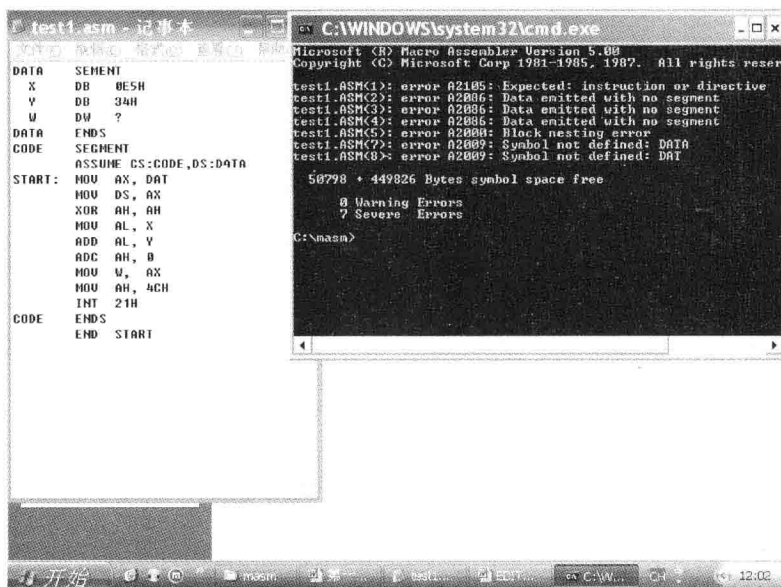


图 1-8 记事本窗口和命令提示符窗口同时出现

应用命令提示符窗口中的 EDIT 程序也能进行程序输入和编辑，但不如记事本简单好用。有关 EDIT 程序操作，请参看 1.5 节介绍。

1.4.4 用 LINK 程序连接目标文件，产生可执行程序

通过编译后，产生了目标文件 TEST1.OBJ。下一步是在命令提示符窗口下，键入“LINK TEST1;”并回车（注意“;”号不能省！由于连接的程序必须是 .OBJ 文件，文件扩展名可以省略）。

系统显示 LINK: warning L4021: no stack segment (LINK 警告 L4021: 没有设置堆栈段)。

表明用户程序没有设置堆栈段，将由系统自动设置。此警告一般不会影响可执行程序的产生和执行。这样，通过连接，产生了 test1.exe 文件。如果要验证，可在命令提示符窗口下输入 DIR 命令，会看到相关文件清单（见图 1-9）。

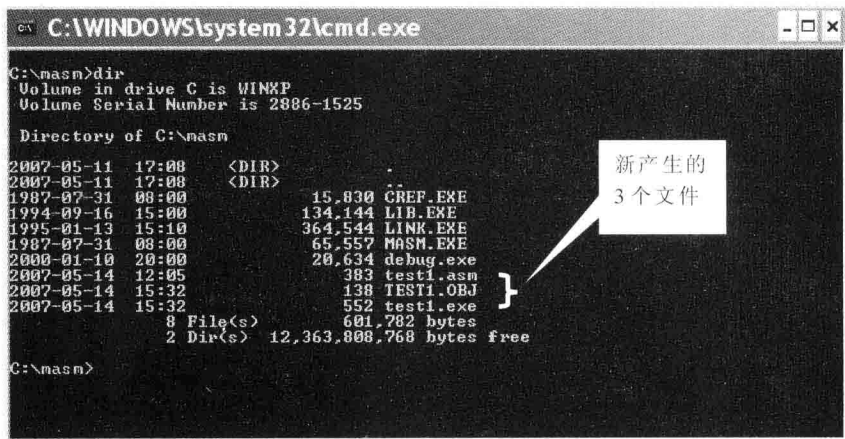


图 1-9 用 DIR 命令了解新增加文件