

全国计算机等级考试

2006

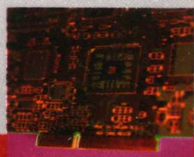


上机题

分类精解与应试策略

三级汇编语言程序设计

全国计算机等级考试命题研究组 编



南开大学出版社

全国计算机等级考试

上机题分类精解与应试策略

三级汇编语言程序设计

全国计算机等级考试命题研究组 编

南开大学出版社

天 津

内容提要

本书主要内容有：① 第1章和第2章，介绍汇编语言上机环境和汇编基础知识，内容涉及80X86宏汇编语言程序设计的基本技术、编程和调试，以及各种常用指令、伪指令、语法等；② 第3章~第7章，将题库中的所有题目划分为数值计算、位运算、编码转换、数据处理和综合题这几大类，通过知识点、题目的分类精解以及举一反三的相关练习题，进行细致透彻的讲解。③ 应试策略，详细列出了备考本科目所应具备的知识，给出复习和考试的秘籍以及注意事项，并说明了上机考试的过程。④ 配套光盘给出本书所有题目的源代码，并提供了上机考试模拟系统，供考生考前练习，可检验实际水平。

本书完全针对准备参加全国计算机等级考试三级PC技术上机考试（三级汇编语言）的考生，同时也可以作为普通高校、大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

另，备考三级PC技术笔试考试的考生，可选购《笔试题分类精解与应试策略 三级PC技术》。

图书在版编目(CIP)数据

全国计算机等级考试上机题分类精解与应试策略. 三级汇编语言程序设计 / 全国计算机等级考试命题研究组编. —天津：南开大学出版社，2006.3

ISBN 7-310-02499-0

I. 全... II. 全... III. ①电子计算机—水平考试—自学参考资料②汇编语言—程序设计—水平考试—自学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字(2005)第 156347 号

版权所有 侵权必究

南开大学出版社出版发行

出版人：肖占鹏

地址：天津市南开区卫津路 94 号 邮政编码：300071

营销部电话：(022)23508339 23500755

营销部传真：(022)23508542 邮购部电话：(022)23502200

*

河北昌黎太阳红彩色印刷有限责任公司印刷

全国各地新华书店经销

*

2006 年 3 月第 1 版 2006 年 3 月第 1 次印刷

787×1092 毫米 16 开本 14.375 印张 351 千字

定价：29.00 元

如遇图书印装质量问题，请与本社营销部联系调换，电话：(022)23507125

前 言

教育部考试中心推出的全国计算机等级考试,是国内除升学考试之外参与人数最多的考试之一,具有相当的权威性、科学性和公平性,它于1994年推出,历经10年发展,已成为我国普及计算机教育不可或缺的组成部分,到目前为止,该考试已经举行过20余次,考生累计已逾千万。计算机等级考试的考试大纲,根据科学技术的发展和社会需求的变化,对全国计算机等级考试的科目、考核内容和考试形式多次进行调整,2004年,推出了等级考试的第5个考试大纲,即2004年版的《考试大纲》。

本书根据教育部考试中心制定的最新考试大纲要求编写,书中覆盖了该门课程在大纲中所提到的所有内容。我们在编写本书的同时,考虑到考生参加等级考试的需要,把各种题型和训练融会在每本书之中,以期给考生带来切实的帮助。

本书的主要特点是针对性强。我们认为,在考试辅导书中,面面俱到并非是一个优势,针对性强才真正对考生有益。本书只针对等级考试三级汇编语言上机考试,不涉及无关内容。我们所分析的题目,都取自考试题库。本书主要内容如下:

① 第1章和第2章介绍汇编语言上机环境和汇编基础知识,内容涉及80X86宏汇编语言程序设计的基本技术、编程和调试,以及各种常用指令、伪指令、语法等,方便考生在复习过程中随时方便地查阅,提高学习效率。

② 第3章~第7章,将所有汇编语言题库中的题目划分为数值计算、位运算、编码转换、数据处理和综合题这几类,分门别类进行详细的讲解。每一章的构成大致有3个方面:知识点部分,这是基础,是理解题意和正确作答的关键;“分类精解”针对题库中的典型题目进行细致透彻的解答分析,由此,考生遇到同类问题,便可以迎刃而解;“举一反三”给出相关类型题目的练习题以及答案,让考生对这方面的知识点真正掌握熟练。

③ 最后一章是应试策略,本章内容不多,却是本书的精华所在,这里详细列出了备考本科目考试所应具备的知识,给出复习和考试的秘籍以及注意事项,并说明了上机考试的过程。

④ 配套光盘。配套光盘中给出了本书所有题目的源代码,并提供了上机考试模拟系统,考生可在考前进行实际练习,通过系统的判分和解析,可检验自己的实际水平。

为了保证本书及时面市和内容准确,很多朋友做出了贡献,廖明武、于樊鹏、齐惠颖、任世华、田民、许伟、侯佳宜、何雄、赵晓睿、戴文雅、戴军、黄志雄、李志云、陈安南、李晓春、王春桥、王雷、韦笑、龚亚萍等老师在编写文档、调试程序、排版、查错、预读、光盘制作与测试等工作中加班加点,付出了很多辛苦,在此一并表示感谢!

在学习的过程中,您如有问题或建议,请与我们联系:book_service@126.com,或登录百分网查找信息和寻求帮助:www.baifen100.com。

全国计算机等级考试命题研究组

2005年12月

目 录

第 1 章 汇编语言及程序调试基础	1
1.1 80X86 宏汇编语言运行环境	1
1.1.1 汇编语言的工作环境	1
1.1.2 汇编程序的使用	2
1.2 80X86 宏汇编语言程序的格式	2
1.2.1 汇编语言程序的格式和结构	2
1.2.2 汇编语言程序的语句类型	3
1.2.3 汇编语言的语句格式	5
1.3 80X86 宏汇编语言的使用	7
1.3.1 建立 ASM 源文件	7
1.3.2 用 MASM 程序产生目标文件	8
1.3.3 用 LINK 程序产生 EXE 文件	13
1.3.4 执行程序	14
1.3.5 检查程序执行结果	14
1.4 EXE 文件和 COM 文件	17
1.4.1 EXE 文件的结构	17
1.4.2 COM 文件的结构	17
第 2 章 80X86 汇编语言程序设计	19
2.1 80X86 指令系统	19
2.1.1 寻址方式	19
2.1.2 8086/8088 指令系统	20
2.1.3 80286 增强和扩充的指令	30
2.1.4 80386 增强和扩充的指令	31
2.1.5 80486 和 Pentium 增强和扩充的指令	32
2.2 汇编语言的数据和表达式	34
2.2.1 常量	34
2.2.2 变量	34
2.2.3 标号	35
2.2.4 运算符和表达式	35
2.3 伪指令语句	38
2.3.1 符号定义伪指令	38
2.3.2 变量定义伪指令	39

2.3.3 段定义伪指令	40
2.3.4 过程定义伪指令	41
2.3.5 其他伪指令	42
2.3.6 宏指令	44
2.4 程序设计基础	46
2.4.1 顺序程序设计	46
2.4.2 分支程序设计	46
2.4.3 循环程序设计	46
2.4.4 子程序设计	47
第3章 数值计算题分类精解	50
3.1 数值计算基础	50
3.1.1 数值计算的基本概念	50
3.1.2 数值计算题目应考策略	50
3.2 序列计算	51
3.2.1 序列计算基本原理	51
3.2.2 序列计算的汇编语言实现	51
3.3 分段计算	60
3.3.1 分段计算基本原理	60
3.3.2 分段计算的汇编语言实现	60
3.4 矩阵运算	68
3.4.1 矩阵计算基本原理	68
3.4.2 矩阵计算的汇编语言实现	68
第4章 位运算题分类精解	80
4.1 位运算基础	80
4.1.1 位运算的基本概念	80
4.1.2 位运算题目应考策略	80
4.2 位组合运算	81
4.2.1 位组合运算基本原理	81
4.2.2 位组合运算的汇编语言实现	81
4.3 逻辑运算	89
4.3.1 逻辑运算基本原理	89
4.3.2 逻辑运算的汇编语言实现	89
第5章 编码转换题分类精解	102
5.1 编码转换基础	102
5.1.1 编码转换的基本概念	102
5.1.2 编码转换题目应考策略	102
5.2 字符转换	103



5.2.1 字符转换基本原理	103
5.2.2 字符转换的汇编语言实现	103
5.3 数值转换	117
5.3.1 数值转换基本原理	117
5.3.2 数值转换的汇编语言实现	118
第6章 数据处理题分类精解	132
6.1 数据处理基础	132
6.1.1 数据处理的基本概念	132
6.1.2 数据处理题目应考策略	132
6.2 插入删除	133
6.2.1 插入删除基本原理	133
6.2.2 插入删除的汇编语言实现	133
6.3 排序	142
6.3.1 排序基本原理	142
6.3.2 排序的汇编语言实现	142
6.4 最值计算	149
6.4.1 最值计算基本原理	149
6.4.2 最值计算的汇编语言实现	150
6.5 数据统计	157
6.5.1 数据统计基本原理	157
6.5.2 数据统计的汇编语言实现	158
第7章 综合题分类精解	169
7.1 综合题的基本概念	169
7.2 综合题目应考策略	169
第8章 汇编语言上机考试应试策略	190
8.1 应试知识准备	190
8.1.1 基本汇编指令	190
8.1.2 汇编程序的编制和调试	192
8.1.3 常用算法的汇编语言实现	193
8.2 复习和准备考试	193
8.2.1 复习阶段	193
8.2.2 熟悉考场环境	194
8.2.3 报名参加考试	194
8.2.4 考试日期规定	195
8.3 上机考试注意事项	195
8.3.1 上机考试要求	195
8.3.2 上机考试时间	195

8.3.3 等待评分结果	196
8.3.4 考试分数计算	196
8.3.5 特殊情况处理	196
8.4 上机考试过程说明	196
8.4.1 登录过程	197
8.4.2 考试过程	199
8.4.3 交卷	200
附录 A 参考资料	202
A.1 DOS 功能调用 (INT 21H)	202
A.2 汇编程序出错信息	207
A.3 DEBUG 的主要命令	213
A.3.1 DEBUG 程序的调用	213
A.3.2 DEBUG 的主要命令	213
附录 B 常用字符与 ASCII 代码对照表	219

第 1 章 汇编语言及程序调试基础

在全国计算机等级考试三级 PC 技术的上机考试中，要求考生熟练掌握 80X86 宏汇编语言程序设计的基本技术、编程和调试。本章就汇编语言编程过程进行讲述，内容包括 80X86 宏汇编语言运行环境、程序格式、编辑、编译、连接以及程序的基本调试技术。

本章所提到的汇编语言均指 80X86 宏汇编语言。

1.1 80X86 宏汇编语言运行环境

1.1.1 汇编语言的工作环境

汇编语言的工作环境是指能使汇编程序顺利运行的软件环境和硬件环境两个方面。下面将分别讲述：

1. 硬件环境

80X86 汇编语言程序一般都在 PC 机及其兼容机上运行。因此，机器具有一些基本配置（如 DOS 操作系统和一定的硬盘空间）即可。

2. 软件环境

软件环境是指支持汇编语言程序运行和帮助建立汇编语言源程序的一些操作系统与软件，包括以下几个部分：

（1）DOS 操作系统

汇编语言程序的建立和运行都是在 DOS 操作系统下进行的。目前，常用的 DOS 操作系统有 MSDOS 和 UCDOS。因此，要对汇编语言程序进行上机操作，首先必须启动 MSDOS 或 UCDOS 操作系统。

（2）文本编辑程序

文本编辑程序是用来输入和建立源程序的一种通用的磁盘文件。程序的修改也是在此状态下进行。

常用的编辑程序有：

- ① 行编辑程序，EDLIN.COM。
- ② 全屏幕编辑程序，EDIT、NOTEPAD、WORDSTAR。

（3）汇编程序

汇编程序有小汇编程序 ASM.EXE 和宏汇编 MASM.EXE 两种。其中，小汇编程序不支持宏汇编，但是宏汇编需要的存储容量大一些，因此，在存储空间足够的情况下，通常选用的是宏汇编 MASM.EXE。

（4）连接程序

80X86 汇编语言使用的连接程序是 LINK.EXE。

(5) 辅助程序

这类程序通常帮助程序员进行程序调试、文件格式的转换等工作。常用的有：

- ① 调试程序 DEBUG.COM。
- ② 转换成 COM 文件格式的 EXE2BIN.EXE 程序。

在某一文件夹中拷入上述程序，则汇编语言的工作环境已经建立，也就是说可以开始进行汇编语言程序的编写和调试工作了（本书中所用的汇编程序均存放在 C 盘的 MASM 文件夹中）。

1.1.2 汇编程序的使用

汇编程序的功能是，对用汇编语言语句编写的不能为机器所识别的源程序进行翻译，把源程序转换成目标文件（即 OBJ 文件）。在转换过程中，汇编程序将对源程序进行两遍扫描，第一遍确定各个变量值，第二遍生成机器代码。如果源程序中有语法错误，则汇编结束后，汇编程序将指出源程序中的错误。程序员可以进入编辑程序，对错误进行修改，然后再次汇编，直到得到没有语法错误的目标文件。

目标文件已经是二进制文件，但它还是不能直接在机器上运行，必须经过连接程序（LINK）把目标文件与库文件和其他的目标文件连接在一起生成可执行文件（EXE 文件），这个文件可以直接在机器上运行。

所以，在计算机上运行汇编语言程序的过程是：

- ① 用编辑程序建立 ASM 源文件。
- ② 用 MASM 程序对 ASM 源文件进行汇编，转换成目标文件。
- ③ 用 LINK 程序进行连接，把 OBJ 文件转换成可执行文件。
- ④ 在 DOS 下直接键入该程序的可执行文件的文件名即可执行该程序。

汇编程序的主要功能是：

- ① 测出源程序中的错误，并给出出错信息。
- ② 生成源程序的目标程序，并给出列表文件（即 LST 文件，该文件列出了汇编语言和机器语言的程序清单）。
- ③ 展开宏指令（即将程序中定义好的宏指令展开为原来的几条语句）。

1.2 80X86 宏汇编语言程序的格式

1.2.1 汇编语言程序的格式和结构

下面给出一个汇编语言程序的例子，以此说明汇编语言程序的格式和结构。

【例 1-1】对数组中所有元素求和并将结果存入 RES 地址单元。

程序 1-1: prog1-1.asm

```
DSEG          SEGMENT                ; 数据段
ARR           DB  32, 45, 16, 75, 89
CNT           DW  $-ARR
RES           DW  ?
DSEG          ENDS
```



```

CSEG          SEGMENT                ; 代码段
               ASSUME CS:CSEG, DS:DSEG
MAIN          PROC    FAR
START:        PUSH    DS              ; 保存返回地址
               SUB     AX, AX
               PUSH    AX
               MOV     AX, DSEG        ; 初始化 DS
               MOV     DS, AX
               MOV     CX, CNT         ; 循环次数送 CX
               LEA     SI, ARR         ; ARR 偏移地址送 SI
               MOV     AX, 0           ; 寄存器 AX 清零
LP:           ADD     AX, [SI]         ; 进行数据相加
               INC     SI
               LOOP    LP
               MOV     RES, AX         ; 保存结果
               RET                    ; 返回 DOS
MAIN          ENDP
CSEG          ENDS
               END      START

```

由此程序可以看出：

① SEGMENT 和 ENDS 把程序分成了很多段，每一段有自己特定的功能。

汇编语言规定，每个程序中最多可以有 4 种类型的段，即代码段、数据段、堆栈段和附加段。每个段在程序中的顺序没有先后，段的数目也没有限制。所以要求在编写程序的时候，要搞清楚需要设置哪些段，各段的内容如何等问题。

在本程序中，一共分了两个段：第一个段为数据段 DSEG，段内存放着原始数据以及程序执行以后的结果；第二个段为代码段 CSEG，段中包含实现程序要求的基本指令。

② 代码段中可以用 PROC 和 FAR 伪指令定义若干过程，这样可使程序的结构更加清晰。同时，把主程序定义为过程，由 DOS 系统调用该过程。进入过程以后，首先把 DS 的内容和 0 作为段地址和偏移地址入栈，以便在程序结束时使用 RET 指令返回 DOS 系统，这是最常用的返回 DOS 的方法。如果在程序开始没有使用把 DS 的内容和 0 作为段地址和偏移地址入栈的语句，则在程序结束时不能直接使用返回指令，而应该调用编号为 4C 中断来返回 DOS；使用方法如下：

```

MOV AX, 4C00H
INT 21H

```

③ 为了便于程序的阅读与修改，在语句的后面适当使用注释，注释以分号“;”开头。

④ 整个源程序必须用 END 语句结束，表示汇编源程序到此为止。END 后面的标号 START 表示该程序的开始执行地址。

1.2.2 汇编语言程序的语句类型

80X86 宏汇编有三种基本的语句，分别是指令语句、伪指令语句和宏指令语句。三种语句各自具有不同的功能，不能互相取代。

1. 指令语句

指令语句是汇编语言程序中使用最多的语句，它由三部分组成：

- ① 一条指令
- ② 它的标号
- ③ 注释。

标号和注释部分可以省略。

汇编程序进行汇编时，一条指令语句被翻译成对应的机器码，对应着机器的一种操作。

例如：

TRANSFER:	LEA	SI, ARR	; 将 ARR 的偏移地址送到 SI 中
标号		指令	注释

这是一条完整的指令语句，它的构成如下：

- ① TRANSFER 为本语句的符号地址，即标号。
- ② 分号后面的内容为本语句的注释，用来说明本语句的功能。
- ③ LEA SI, ARR 是 8086 宏汇编的一条数据传输指令。

2. 伪指令语句

伪指令语句是为汇编程序和连接程序提供一些必要控制的管理性语句。

汇编程序进行汇编时，对伪指令语句不产生目标代码，汇编程序将它的意义解释之后，即完成相对应的伪操作。例如分配存储单元，给符号赋值等。

汇编完成以后的部分信息（即存储单元的分配、符号赋值等）将被保留下来，供连接程序使用。

伪指令语句的组成也是分为三部分，即：

- ① 标号
- ② 伪指令
- ③ 注释。

与指令语句不同的是，它的标号后面没有冒号。例如：

CNT	DW	0345H	; 给变量 CNT 分配一个字存储单元，并且赋值 0345H
标号		伪指令	注释

这是一条完整的伪指令语句，CNT 是它的标号，它代表由伪指令 DW 分配的那个存储单元的符号地址，又可以称为变量名。汇编程序对这条语句进行汇编以后，为 CNT 分配一个字的空间，并将 0345H 存入其中。由于这条伪指令的全部功能在汇编时已经完成，所以在机器代码中，这条语句不会出现。

3. 宏指令语句

与指令语句和伪指令语句一样，宏指令语句也由标号、宏指令和注释组成。宏指令是编程者按一定规则来定义的一种较“宏观”的指令。一般来说，一条宏指令包含很多条指令和伪指令语句，宏指令语句通常在程序需要在不同地方重复某几条语句的情况下使用，把这些重复的语句定义为宏指令后，在后面的程序中凡是出现这几条语句的地方都用宏指



令来代替,这样处理后的源程序书写精练,可读性好。

在汇编时,汇编程序按照宏指令的定义,在出现宏指令的地方将其展开还原。因此,从源程序来看,利用宏指令节省了篇幅,且使程序变得简单,但这并不代表程序的目标代码文件缩小。

1.2.3 汇编语言的语句格式

汇编语言程序中的每个语句可由四个字段构成,格式如下:

```
[NAME] OPERATION OPERAND [; COMMENT]
```

其中:

- ① 名字字段 (NAME) 是一个符号。
- ② 操作符字段 (OPERATION) 是一个操作码的助记符,它可以是指令、伪指令或者宏指令。
- ③ 操作数字段 (OPERAND) 由一个或多个表达式组成,它提供为执行操作符字段指令所需要的信息。
- ④ 注释字段 (COMMENT) 用来说明程序或语句的功能。分号表示注释的开始,也将注释项与前面的部分分开。注释也可以从一行的第一个字符开始,这时这一行都是注释,这种注释通常用来说明某一个程序段的功能。

汇编语言的语句格式中,带方括号的字段是可以省略的,各字段之间需要用空格分开。下面讲述各字段的表示方法及注意事项。

1. 名字字段

在源程序中,名字用下列字符来组成:

- ① 字母: A~Z
- ② 数字: 0~9
- ③ 专用字符: ?、.、@、_、\$

由以上字符组成的名字必须满足如下规则:

除数字外,所有的字符都可以放在名字的第一个位置。

如果名字中用到“.”字符,必须放在第一个位置。

名字最长不能超过 31 个字符,汇编程序对 31 个字符以后的字符不予处理。因此,如果两个超过 31 个字符的名字,其前 31 个字符相同,则汇编程序就会将它们当成同一个名字,从而导致错误,这一点需要注意。

一般来说,名字可以是标号或变量。它们都是用来表示本语句的符号地址,是可有可无的,只有当需要用符号地址来访问该语句时才必须出现。

① 标号: 标号在代码段中定义,后面要跟冒号,它可以用 LABEL 或 EQU 伪操作来定义。此外,它还能用来作为过程名。由于标号和一条语句的地址相联系,因此经常用在转移指令或 CALL 指令的操作数字段上。

② 变量: 变量在除代码段以外的其他段中定义,后面不跟冒号。也可以用 LABEL 或 EQU 伪操作来定义,变量经常出现在操作数字段。

变量和标号都具有三种属性:段属性、偏移属性和类型属性。

① 段属性。这个属性定义了标号和变量的段起始地址，它必须在一个段寄存器中。标号的段是它所在出现代码段，所以总是在 CS 寄存器中。变量的段可以在 DS、ES、SS 和 CS 中，但通常是在 DS 和 ES 中。

② 偏移属性。标号和变量的偏移地址都是一个 16 位的无符号数，它代表从段起始地址到定义标号的位置之间的字节数。对于变量来说，在当前段内的偏移值等于当前地址计数器的值，当前地址计数器的值可以用 \$ 来表示。

③ 类型属性。对于标号而言，类型属性用来指出该标号是在本段内引用还是在其他段内引用。如果在本段内引用，类型为 NEAR，指针长度为 2 字节；如果在其他段内引用，类型为 FAR，指针长度为 4 字节。对于变量来说，其类型属性定义该变量所占用的字节数。这个属性由定义变量的伪指令 DB、DW 或 DD 确定。其中，DB 把变量定义为字节型，DW 把变量定义为字型，DD 把变量定义为双字型。其类型分别为 BYTE、WORD 和 DWORD。

2. 操作符字段

操作符可以是硬指令、伪指令或宏指令的助记符。硬指令是指指令系统中的指令，汇编程序将其翻译成对应的机器码。伪指令不能翻译成对应的机器码，它只是在汇编过程中完成相应的控制操作，所以又称为汇编控制命令。宏指令是给一组有限的指令（硬指令、伪指令）定义的代号，汇编程序在对源程序进行汇编时将根据其定义展开成相应的指令。

3. 操作数字段

操作数字段是操作符的操作对象，操作符完成相应的操作要求有一系列的操作数。操作数字段由一个或多个表达式组成，多个操作数字段之间一般用逗号分开。对于指令，操作数字段一般给出操作数地址，它们可能有一个或两个，或者一个也没有。对于伪操作或宏指令，则给出它们所要求的参数。

操作数通常可以是常数、寄存器、存储器操作数和表达式。

(1) 常数

常数是没属性的数，它在汇编时已经完全确定，程序运行中不会发生改变。8086 宏汇编中，允许使用以下几种形式的常数：

① 二进制常数：二进制常数是一串 0 和 1 数字的组合，数字的末尾以 B 结束。

② 十进制常数：十进制常数是最常用的一种常数，后面跟字母 D 或者不跟任何字母。

③ 十六进制常数：十六进制常数由数字 0~9 和字母 A~F 组成，这类常数必须以字母 H 结尾，以数字 0~9 开头。例如：数字 FE34H 的正确写法应该是 0FE34H，尽管这两个数的数值相同，但前面的那个可能被汇编程序识别为标识符。

④ 八进制常数：八进制常数由数字 0~7 组成，逢八进一。

⑤ 串常数：用单引号括起来的字符及字符串，其各个字符的 ASCII 码值构成串常数。例如：123 的值是 313233H，而不是 123；BC43H 的值为 42433433H。所以，串常数可以和正数常数等价使用，但这些串常数的长度必须为一个字节或一个字，这样才能与目标操作数的长度相匹配。

(2) 存储器操作数

存储器操作数包括标号和变量，它可以用作源操作数也可以用作目的操作数，但不能同时充当源操作数和目的操作数。



变量通常是指存放在某些存储单元中的数据,这些数据在程序的运行过程中可以改变。变量通过标识符来引用,可以采用直接寻址、基址寻址和变址寻址方式进行存取。

标号是可执行的指令性语句的符号地址,可以用来作条件转移指令和子程序调用指令的目标操作数。

(3) 表达式

表达式是由常数、寄存器、标号、变量和一些运算符相组合而成,可以有数字表达式和地址表达式两种。

1.3 80X86 宏汇编语言的使用

1.3.1 建立 ASM 源文件

下面通过一个例子来具体说明汇编语言程序上机运行的过程。

【例 1-2】数据段中有一个字数组 ARR,数组的第一个字中存放着该数组的长度。编写一个程序,使该数组中的数按照从小到大的次序排列。

使用 EDIT 程序编写源文件,在 DOS 环境下调用 EDIT。

C:\>EDIT✓

其中,“✓”代表回车。然后在该环境下进行程序的编辑,建立 prog1-2.asm 的源程序,如下所示:

```

DSEG      SEGMENT                ; define data segment
COUNT    DW  ?
ARR        DW  6, 34, 63, 14, 53, 94, 26
DSEG      ENDS                    ; end of data segment
CSEG      SEGMENT                ; define code segment
ASSUME CS:CSEG, DS:DSEG
MAIN      PROC FAR                ; main part of program
START:     PUSH  DS                ; save old data segment
           MOV   AX, 0              ; put zero in AX
           PUSH  AX                ; save it on stack
           ; set DS register to current data segment
           MOV   AX, DSEG           ; data segment address
           MOV   DS, AX            ; into DS register
           MOV   SI, OFFSET ARR     ; put offset arr of source
                                   ; buffer in SI
           MOV   CX, [SI]           ; put number of arr into CX
           MOV   COUNT, CX          ; transfer CX to COUNT
AGAIN1:    MOV   BX, 1              ; set flag (BX)=1
           DEC   COUNT              ; get ready for COUNT-1 compares
           JZ    EXIT              ; exit if COUNT is 0
           LEA   SI, ARR
           MOV   CX, COUNT          ; load COUNT into CX
AGAIN2:    ADD   SI, 2              ; address a data el.
           MOV   AX, [SI]           ; load it into AX

```

```

CMP     [SI+2], AX      ; is next el.<this el.?
JAE     NEXT            ; no, go check next pair
XCHG    [SI+2], AX      ; yes, exchange these elements
MOV     [SI], AX
MOV     BX, 0           ; set exchange flag 0
NEXT:    LOOP    AGAIN2
        CMP     BX, 0    ; any changes made?
        JE      AGAIN1  ; yes, process list again
EXIT:    MOV     SI, OFFSET ARR ; no, restore registers
        RET              ; return to DOS
MAIN     ENDP          ; end of main part or program
CSEG     ENDS          ; end of code segment
END      START         ; end assembly

```

1.3.2 用 MASM 程序产生目标文件

源文件建立以后,就要使用汇编程序对源文件进行汇编,汇编以后产生二进制目标文件 (OBJ 文件)。

汇编之前应该注意,源程序文件应该和汇编程序在同一个文件夹,或者在进行汇编的时候给出源程序文件所在的路径。如果汇编程序和源程序文件不在同一个文件夹并且没给出路径的话,汇编将会发生错误。比如汇编程序在 D 盘的 MASM 文件夹中,而源程序文件直接存在 D 盘的 ASM 文件夹中,则直接使用 `masm PROG1-2.asm` 进行汇编时,就会出现不能打开源文件的错误,如图 1-1 所示。

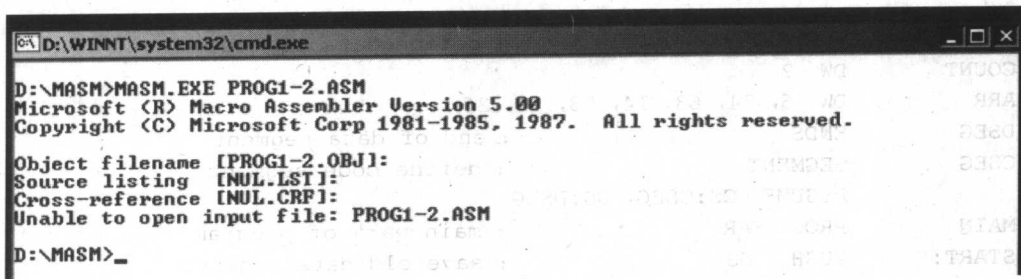


图 1-1 错误提示

为了避免这种错误的发生,可以在进入汇编程序所在的文件夹以后再执行 `EDIT` 命令,参见图 1-2。

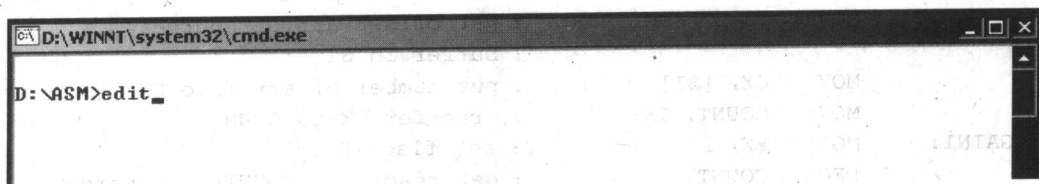
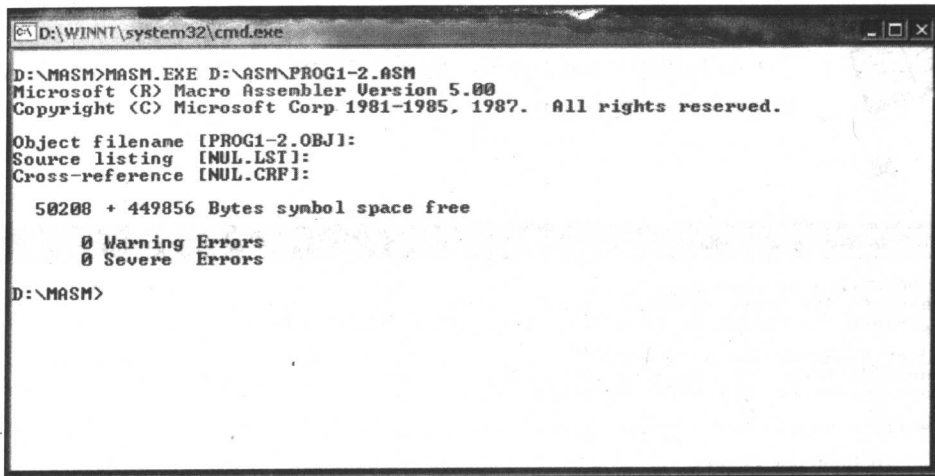


图 1-2 进入汇编程序所在的文件夹以后再执行 EDIT 命令

这样,系统就会默认为将文件存到 MASM 文件夹中(本书后面的讲解都是在 MASM 文件夹中进行)。或者给出源程序文件所在的路径,即用: `D:\MASM>masm D:\asm\progl-2.asm` 进行汇编,路径正确以后汇编的结果如图 1-3 所示。



```
D:\WINNT\system32\cmd.exe
D:\MASM>MASM.EXE D:\ASM\PROG1-2.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [PROG1-2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50208 + 449856 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\MASM>
```

图 1-3 正确的结果

以上信息中，方括号中的信息为该项目的缺省回答，若不改变缺省值，则直接按回车键略过。

源程序进行汇编以后，将产生以下三个文件：

① OBJ 文件：这个文件是汇编后生成的目标代码文件，是进行汇编产生的主要文件，任何一次汇编都应该生成以 .OBJ 为扩展名的文件。我们只要在“Object filename [prog1-2.OBJ]:”后直接回车，就会生成 prog1-2.obj 文件。



注 意

当源程序有错的时候，不会生成这个文件。因此，应该特别注意上述信息中给出的 Warning errors 和 Severe errors 的类型和数目，当 Severe errors 数目不为 0 的时候，不会生成 OBJ 文件。当 Warning errors 数目不为 0 的时候，可以生成 OBJ 文件，如果编程者可以容忍这些错误的话，可以继续下进行下面的步骤。

② LST 文件：这个文件是列表文件，用来列出汇编语言和机器语言的程序清单，并给出符号表。正确使用该文件可以使程序的调试更加方便。如果不需要这个文件，则在“Source listing [NUL.LST]:”后面直接按回车即可；如果需要，则要在“Source listing [NUL.LST]:”后面键入文件名。上面的汇编过程没有生成 LST 文件，为了更清楚地了解汇编过程，将在后面对 LST 文件进行说明。

③ CRF 文件：这个文件是用来产生交叉引用表 REF 的，交叉引用表给出了用户定义的所有符号，对于每个符号列出了其定义所在的行号（加上#）及引用的行号。对于一般的程序不需要建立此文件，所以在“Cross-reference [NUL.CRF]:”后面键入回车即可；如果需要建立交叉引用表的话，则应该在“Cross-reference [NUL.CRF]:”的后面键入文件名。上面的汇编过程同样没有生成 CRF 文件，对它的说明也将在下面进行。

下面将对 prog1-2.asm 进行汇编，在汇编过程中需要生成 LST 和 CRF 文件，并且要建立交叉引用表，其操作与汇编的过程如图 1-4 所示。