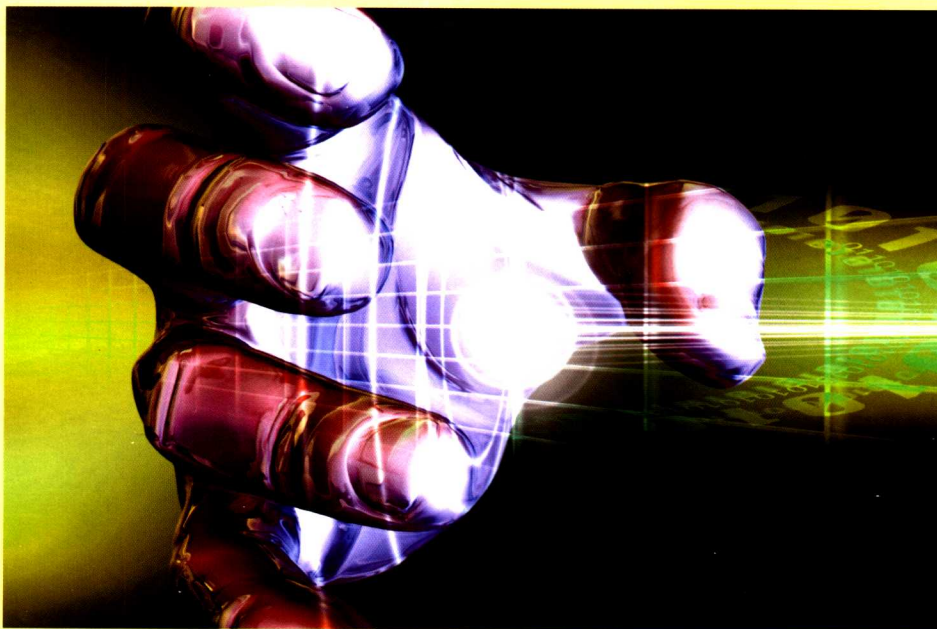


微机原理与接口技术

课程设计

宋杰 汪志宏 江敏 编著



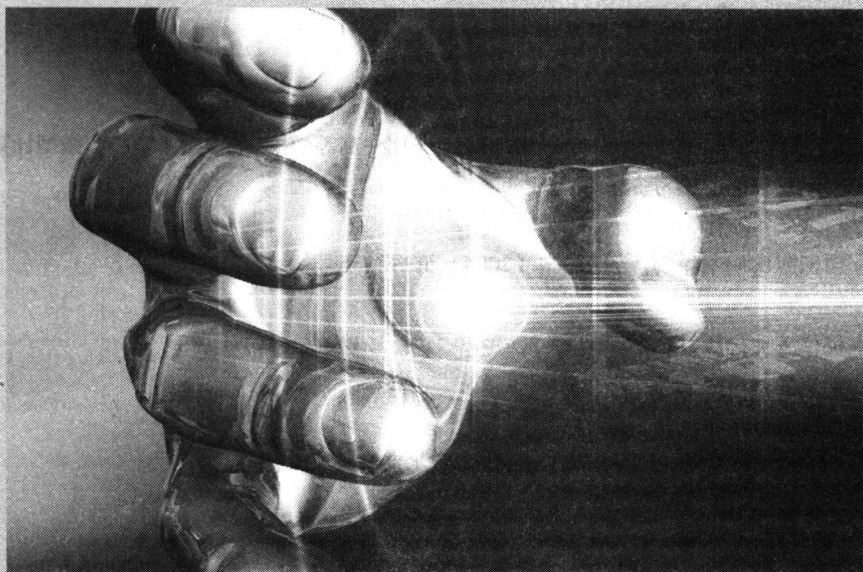
机械工业出版社
China Machine Press

高等院校计算机课程设计指导丛书

微机原理与接口技术

课程设计

宋杰 汪志宏 江敏 编著



机械工业出版社
China Machine Press

本书是微机原理和接口技术等课程的课程设计指导书, 可以作为“微机原理和接口技术”、“汇编语言程序设计”等课程的实验和辅助教材。

本书结合教学内容, 从应用的角度出发, 提供了27个小型的课程设计。这些课程设计分为三类: 汇编语言程序设计、微机标准接口课程设计和接口技术课程设计。此外, 本书还提供了两个复杂的课程设计供读者学习、研究。每个课程设计都给出了设计目的、要求及设计原理和思路, 方便读者顺利进行设计。附录中提供了各个项目的源程序以供参考。

本书可作为高等学校计算机、电子和自动化等相关专业学生的课程设计教材, 也可供从事微机应用的工程技术人员参考。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

微机原理与接口技术课程设计/宋杰等编著. —北京: 机械工业出版社, 2005.4
(高等院校计算机课程设计指导丛书)
ISBN 7-111-15943-8

I. 微... II. 宋... III. ①微型计算机-理论-高等学校-教学参考资料②微型计算机-接口-高等学校-教学参考资料 IV. TP36

中国版本图书馆CIP数据核字(2004)第142135号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

策划编辑: 温莉芳

责任编辑: 朱 劼

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2005年4月第1版第1次印刷

787mm×1092mm 1/16·14.75印张

印数: 0 001-5000册

定价: 23.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

高等院校 计算机课程设计指导丛书

专家指导委员会

(以姓氏拼音为序)

陈向群	(北京大学)
戴葵	(国防科技大学)
何钦铭	(浙江大学)
林闯	(清华大学)
刘振安	(中国科技大学)
马殿富	(北京航空航天大学)
齐勇	(西安交通大学)
宋方敏	(南京大学)
汤庸	(中山大学)
王立福	(北京大学)
吴功宜	(南开大学)
赵一鸣	(复旦大学)

联络人 温莉芳

丛书序言

近年来,我国在计算机应用、计算机软件和电子类相关专业的人才培养方面,取得了长足的进展,每年的毕业生都有数十万人。但是这些毕业生走进企业、公司、政府机构或研究单位之后,往往深刻地感觉到缺乏实际开发设计项目的经验,不善于综合运用所学理论,对知识的把握缺乏融会贯通的能力。

综合考察目前高等院校教学大纲、课程设置以及内容安排等方面的情况,多数学校还是比较重视训练学生的实际设计能力。但是,从安排设计实践的内容上看,基本上是围绕相关课程教学内容而展开的,不能够构成对实际问题的解决方案;从配套程序的规模上看,一般只是几十行到几百行的源代码,或者是一个单独电路的设计,远远小于一个小型项目的规模;从设计的结构上看,由于设计实践是围绕着课程教学内容而进行的,问题已经高度抽象,学生很难得到有关综合运用所学知识的整体训练机会。而且,这些内容相对简单、问题域已经高度抽象、规模较小的设计实践一人基本上就能完成,学生几乎无法通过这些设计实践,去真正获得有关项目管理和团队协作等方面的基本训练和工作经验。

由此可以看出,大多数学校对学生实际设计能力的训练与国外知名大学和国内精品课程相比较,还是存在一些差距的。为此,机械工业出版社华章分社和一批高等院校的教师,针对当前高等院校计算机硬件、软件和电子类相关课程教学中存在的问题,参考国内外知名大学相关课程成功的教学经验,设计编写了这套“高等院校计算机课程设计指导丛书”,其目的就是通过课程设计的一系列训练,把知识获取和项目实践两个方面有机地结合起来。

在这套“高等院校计算机课程设计指导丛书”中的每一门课程设计里,都安排了由多个子项目组成的一个课程设计项目。学生们可以在教师的指导下,逐步设计实现这些子项目,并最终完成一个功能相对完整,可以运行的系统,其代码可以是数千行,甚至上万行。通过这种设计课程,学生一方面可以结合课程的教学内容循序渐进地进行设计方面的实践训练,另一方面,在参与一系列子项目的实践过程中,还能提高如何综合运用所学知识解决实际问题的能力,以及获得有关项目管理和团队合作等等众多方面的具体经验,增强对相关课程具体内容的理解和掌握能力,培养对整体课程知识综合运用和融会贯通能力。

参加丛书编写的各高等院校的教师都有着丰富的教学、科研,以及与企业合作开发项目等多方面的经验。每个课程设计中的子项目和整体项目,都来自教师们具体的科研和设计开发实践,所选设计项目与教学内容配合紧密,项目的难度与规模适宜。

最后,感谢机械工业出版社华章分社编辑们的大力支持,使出版有关这套丛书的计划,从单纯的构想演化成带有油墨芳香的真实。

丛书写作组

前 言

微机原理和接口技术是计算机和电子等专业重要的基础专业课程之一，不但要求有较高的理论水平，而且还要求有实际的动手能力。

本课程设计的主要目的是提高实践能力，包括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，从而学以致用。多年的课程教学和实验教学的实践证明，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，从中得到收益和提高。

在实验类别上，本书尝试了新的分类方法。本书将实验划分为三种类型。汇编程序设计部分是在学习所有微机原理和接口技术课程时必须掌握的，微机内部标准接口设计提供了已集成在微机内部的标准接口的编程和应用。这两部分在所有的微机上都很容易实现，不用再添加硬件，因此便于在所有的学校实施。第三部分是基于实验平台的接口实验，提供了常见的接口课程设计。对于接口电路，只有多做实验才能真正达到灵活应用的目的。

本书融入了作者多年的教学经验，在内容选择上既保留了一些经典的设计内容，也开发了一些新的设计项目，注重实用。在内容层次、语言表达上以及概念描述上力求通俗易懂，突出主要概念和知识点。

本书包括四章和五个附录。

第1章概括介绍全书特点和内容安排。

第2章为课程设计基础，介绍汇编语言的编译过程和常用软件，着重描述了BIOS和DOS的中断功能调用，并介绍了本课程设计的所需的软硬件环境。

第3章为课程设计项目。该章分为三个部分，第一部分为汇编语言练习，希望通过一系列的课程设计，使读者熟悉汇编语言的基本方法。第二部分为微机系统自身所含接口的控制设计。前两部分课程设计基本上在各种档次的机器中都可实现，不受机器的影响。第三部分是机外接口实验，这一部分和所采用的实验系统有一定的联系。但本书所选用的实验在几种主流实验台上都可实现，只是这几种系统的端口地址不一致，但只要在实现时修改相应地址即可。

第4章提供了两个复杂的设计以供分析、研究。

附录中给出了所有课程设计的源程序及ASCII码表、键盘扫描码表、DOS和BIOS功能调用表，可供读者编程时参考使用。

本书在课程设计过程中对原理也尽可能给予较详细的叙述，同时介绍了各种课程设计方法，给出了大量的图表，以便读者理解设计内容。此外，还给出了参考程序的流程图和硬件实验的接线图以便于实验。

通过完成本书提供的课程设计，读者可以掌握微型计算机系统的工作原理，培养初步的硬件接口的工程设计能力。

书中所有程序都在微机和实验平台上调试通过，读者可登录华章网站 (<http://www.hzbook.com>)

或<http://www.ahu.edu.cn/~istp/asm/download.htm>下载所有源程序。

许多人对本书的编写工作给予了帮助，在此一并表示感谢，特别感谢刘振安教授给予的关心和指导。

本书由安徽大学宋杰主笔，汪志宏、江敏承担了部分程序编制和绘图的工作。

由于计算机技术发展日新月异，新技术层出不穷，我们在编写过程中虽已尽了最大的努力，但错误和不妥之处在所难免，恳请广大读者指正。

编 者

2004年10月

目 录

专家指导委员会	
丛书序言	
前言	
第1章 课程设计概述	1
1.1 课程设计的必要性	1
1.2 课程设计目标	1
1.3 课程设计的总体评价方式	2
第2章 汇编语言实验基础	5
2.1 汇编语言的开发过程	5
2.2 汇编语言常用软件的使用方法	6
2.2.1 宏汇编程序MASM	6
2.2.2 调试程序DEBUG	8
2.3 BIOS中断和DOS中断	15
2.4 课程设计环境	16
第3章 简单项目的课程设计	17
3.1 汇编语言部分	17
3.1.1 闰年计算	17
3.1.2 系统驻留时间程序	18
3.1.3 代码转换	21
3.1.4 输入输出显示	23
3.1.5 BCD码相乘	24
3.1.6 计算N的阶乘	25
3.1.7 打字计时练习	25
3.1.8 光条式程序菜单设计	27
3.1.9 写文件	30
3.2 微机内部接口部分	34
3.2.1 微机串口通信内环自检	34
3.2.2 微机串口通信功能综合检测	40
3.2.3 音乐演奏	42
3.2.4 硬件中断时钟设计	47
3.2.5 鼠标接口	47
3.2.6 打印接口	51
3.3 系统接口部分	54
3.3.1 串行通信	55
3.3.2 红绿灯设计	61
3.3.3 小键盘按键识别数码管移位显示	65
3.3.4 8259A中断控制器实验	68
3.3.5 RAM实验	70
3.3.6 DMA数据传输	72
3.3.7 定时器/计数器实验	75
3.3.8 数模转换	77
3.3.9 模数转换	80
3.3.10 LED显示设计	84
3.3.11 PWM脉宽调制	86
3.3.12 步进电机实验	89
第4章 复杂项目的课程设计	93
4.1 简易文本编辑器	93
4.2 学生成绩管理程序	120
附录A 参考源程序	139
附录B ASCII字符表(7位码)	211
附录C 101、102和104键键盘扫描码	213
附录D DOS中断功能调用表	215
附录E BIOS中断功能调用表	221
参考文献	225



第1章

课程设计概述

微机原理和接口技术是计算机及电子类专业重要的基础专业课之一，它不但要求有较高的理论水平，而且还要求有较强的实际动手能力。本课程的主要目的是提高实践能力，包括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。

与其他课程（如语言类课程设计）不同的是，本课程设计对硬件有很高的依赖性。因此，课程设计的灵活性受到一定的限制，必须有相应的实验平台支持。国内很多高校和公司也开发了不同的实验平台，所幸的是平台主要集中在几个实验机型号上，所采用的芯片和电路也基本相同。目前广泛应用的实验系统主要有清华大学的TPC-1实验系统，TPC-H的实验系统，复旦/启东的实验系统，以及西安唐都的实验系统等。几种系统各有特点，各高校选择的也不尽相同。好在其基本原理都是一致的，因此，使用者可以根据所用实验设备和教材选择相应的课程设计内容。

1.1 课程设计的必要性

长期以来，注重书本知识，轻视实际动手设计能力是教学中普遍存在的现象。进行课程设计的目的就是要改变这种状况，使学习者既具有完备的理论知识，也能够解决在实际学习、生活工作中遇到的问题。学生应该能够把学到的理论知识用到实际中去，将知识转变为实际的生产力，同时通过这些理论知识和实际的应用推动理论和实践的发展。

课程设计能把课本中分布学习的知识，在设计中综合加以应用，进而得到巩固、加深和发展。同时，学习者通过学习查找、运用设计资料，完成工程设计所必备的基本训练。

通过课程设计，能够对所学的知识有更进一步的理解，并能掌握学习理论时没有注意的细节。在实际工作中，只要有一个环节没有做好，整个体系都不能正常运行。调试程序、排除故障有助于提高分析问题、解决问题的能力。课程设计中碰到的挫折，有助于养成良好的学习习惯、严谨的工作作风。

1.2 课程设计目标

一般来讲，课程设计比教学实验复杂一些，涉及的深度更广并更加接近实用。目的是通过课程设计的综合训练，培养学生实际分析问题、编程和动手能力，最终帮助学生系统掌握该门课程的主要内容，更好地完成教学任务。

本课程设计的特点是：

1) 本课程设计分为三个部分:

- 汇编语言练习部分

通过这部分的练习, 学生要熟悉汇编语言的寻址方式和指令系统, BIOS和DOS系统功能调用, 程序设计方法并熟悉宏汇编的编程环境。这部分设计内容可供所有的课程使用。

- 微机自身接口部分

此部分的项目和硬件相关, 但使用的是微机自身的标准硬件和接口。这部分设计内容可供所有的课程使用。

- 实验系统接口部分

因为本部分的内容要使用实验系统来完成, 所以和实验系统有一定的联系, 但本部分内容在各个系统上基本上都可实现, 只是具体方法上可能略有区别。

2) 配合实际应用的要求, 本课程设计既覆盖知识点, 又接近工程实际需要。通过激发学习兴趣, 调动学生主动学习的积极性, 并引导他们根据实际编程要求, 训练他们实际分析问题的能力 & 编程能力, 并养成良好的编程习惯。

3) 通过详细的实例, 循序渐进地启发学生完成设计。课程设计将要求、算法和源程序分开, 为学生创造独立思考的条件。学生在全面理解要求和算法的前提下, 完全可以不按书中提供的参考程序, 设计自己的应用程序。

4) 有些课程设计提出了一些改进措施或要求, 有兴趣的学生可以用来扩充自己的设计。

5) 在内容取舍上, 强调掌握基本的接口技术, 保留了一些经典的内容, 同时将一些不常用的内容适当减少或取消, 如低分辨率的显示模式编程, 但相应也增加了一些如鼠标控制等新内容。

6) 后续的课程设计尽量引用前面的课程设计内容, 以便增加印象并加深理解。

7) 对于同一类型的实验, 提出了不同的实现方法, 供学习实践, 以满足不同学校和不同学生的要求。

另外, 在实际编程中, 为了提高编程质量, 对空行、空隔和注释均有要求。本书也尽可能地根据实际编程要求给出空行、空隔和注释, 有时因为标题和页码等实际原因, 也会适当减少空行、空隔和注释, 但希望学生在书写代码时, 还是应该严格要求处理, 以便建立良好的编程风格。

1.3 课程设计的总体评价方式

由于本课程的设计内容不同, 很难有一个固定的评价标准。根据所做的设计, 评价体系的重点可有所改变。

课程设计的第一部分主要考察学生对汇编语言的掌握, 包括对编译, 链接等编程操作的掌握。

课程设计的第二部分主要考察对微机自身组成的了解程度, 要求对中断等知识有很好的理解并能熟练地应用。

课程设计的第三部分主要考察对常用外部接口的掌握程度。这部分要求不仅对汇编语言,

而且对接口硬件电路都必须很好的掌握。

建议课程设计的结果按优、良、及格和不及格来评价打分。

- **优** 对设计任务理解透彻、能够全面、正确地完成设计内容所规定的任务，并严格按设计要求完成任务。能够写出准确、规范的设计报告。如果对设计要求能进一步优化和提高，或者独立用不同的思路完成设计要求，有创新，也可考虑给优。
- **良** 按设计任务要求能中规中矩地完成设计，程序、电路都能按要求正确工作，设计报告符合要求。
- **及格** 基本完成设计的目标，可能有若干小的缺陷，在帮助下能够完成设计任务。
- **不及格** 不能完成指定设计任务者。

教师可以根据每个项目的不同要求对学生进行评价。



第2章 汇编语言实验基础

2.1 汇编语言的开发过程

汇编语言程序的开发过程如图2-1所示。这个过程主要由编辑、汇编、链接等几个步骤构成。

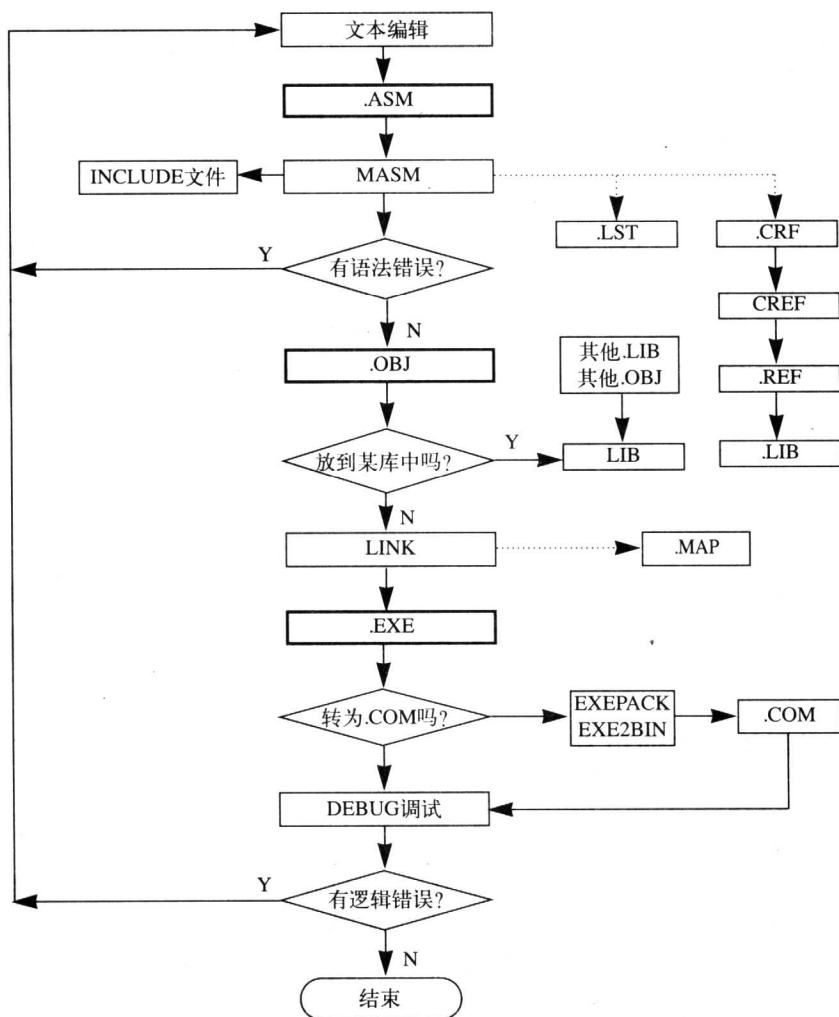


图2-1 汇编语言源程序的建立、编译和链接过程

1. 源程序的编辑

编辑就是调用编辑程序把源程序输入内存, 生成一个扩展名为ASM的文本源文件并存入磁盘。如果是对原有的.ASM文件进行修改, 还要生成一个扩展名为BAK的备份文件, 它是修改前的.ASM文件自动改名生成的。DOS提供的EDIT.EXE或其他全屏幕编辑软件都能完成编辑任务。也可利用Windows的友好界面, 采用文本编译器, 先编好源程序, 再汇编。

2. 源程序的汇编

汇编就是调用汇编程序(如MASM或TASM)对源程序进行翻译, 生成扩展名为OBJ的目标文件。在汇编过程中, 若汇编器检查到源程序中有语法错误, 则不生成目标文件, 并给出错误信息。根据用户需要, 汇编程序还可生成列表文件(.LST文件)和交叉引用文件(.CRF文件)。

3. 目标程序的链接

链接的过程是: 调用链接程序(如LINK或TLINK)将用户目标程序和库文件进行链接、定位, 生成扩展名为.EXE的可执行文件。链接时, 如果在目标文件或库文件中找不到所需的链接信息, 则链接程序给出错误信息, 而不生成可执行文件。根据用户需要, 链接程序还可以生成内存分配文件(.MAP文件)。

4. .COM文件的生成

按照.COM文件的汇编格式设计的源程序, 在生成.EXE文件之后, 还要转换成.COM文件, 如果链接时使用TLINK.EXE程序, 在TLINK命令后加选项“/t”, 可直接生成.COM文件。但是宏汇编MASK.EXE(5.0版本以下)不支持此项功能, 还需要调用转换程序EXE2BIN.EXE, 才能将.EXE文件转换成.COM文件。假设用户的.EXE文件、.COM文件和转换程序都在同一目录下, 则键入以下代码进行转换:

```
EXE2BIN  文件名.EXE  文件名.COM
```

如果不能转换就给出错误信息。

5. 调试可执行程序

如果可执行程序运行之后没有得到预想的结果, 就要重新审查源程序, 找出算法或者逻辑错误, 再重新编辑、汇编、链接和执行。在调试程序的时候, 也可以借助调试软件来进行调试。

2.2 汇编语言常用软件的使用方法

2.2.1 宏汇编程序MASM

要用汇编语言产生在微机上可运行的可执行程序, 如.EXE文件、.COM文件, 先要建立源文件, 然后对源文件进行汇编, 汇编生成的.OBJ文件还不能在微机上直接运行, 必须再通过链接程序将它转为可执行程序。目前汇编语言编译最常用的是宏汇编程序, 主要有汇编程序MASM、链接程序LINK, 以及一些辅助程序如DEBUG调试程序等组成。

下面介绍它们的使用方法。

1. 源文件的建立和汇编

调用全屏幕编辑程序EDIT.EXE、PE.EXE或WS.EXE等,用键盘键入完整的按程序编写格式书写的源程序,退出编辑系统时,磁盘上就建立了一个汇编语言的源程序文件,其扩展名应为ASM。

调用小汇编程序ASM.EXE或宏汇编程序MASM.EXE对源文件进行汇编,汇编程序对.ASM文件进行两遍扫描,汇编后产生二进制目标文件(.OBJ文件)。汇编程序主要有以下功能:

- 1) 检查源程序中的语法错误,给出出错信息。
- 2) 产生目标文件(.OBJ文件)、列表文件(.LST文件)和交叉引用文件(.CRF文件)。
- 3) 展开宏指令。

汇编过程及屏幕显示如下:

```
C: >MASM EXAMPLE      : 源文件名
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved
Source Filename [.ASM]: EXAMPLE
Object Filename [EXAMPLE.OBJ]:
Source Listing [NUL.LST]: (可打入源文件名或省略)
CrossReference CNULL.CRF]: (可打入源文件名或省略)
0      Warning  Errors
0      Severe   Errors
```

汇编程序调入后,先显示版本号,然后提示用户输入源文件名,若用户原来已输入过源文件名,则此行不显示。下面有三个提示行,相应输出三个文件,第一个为.OBJ文件,这是目标文件,是必需的,用户键入回车即可。第二个为.LST文件,是列表文件,若需要列表文件就输入文件名,否则就不产生此文件。第三个为.CRF文件,是建立交叉引用文件,若需要.CRF文件就输入文件名,否则直接回车,不产生此文件。然后,汇编程序对源文件进行汇编,屏幕上显示出源程序中的错误信息,包括错误语句行号、错误代码和错误类型。最后列出警告错误和致命错误的总数。

.LST文件是可打印文件,它同时列出源程序和机器语言清单,并给出行号及符号表,表中给出段名、段的大小及属性,用户定义的符号名、类型及属性等。

.CRF文件用来产生交叉引用表,可以对符号进行前后对照,它给出了用户定义的所有符号,每个符号定义时所在行号以及引用时所在行号。若要打印,就需要调用系统中的CREF程序,运行过程如下:

```
C: > CREF
Cref Filename [.CRF]: <文件名>
List Filename [文件名.REF]:
```

建立了一个.REF文件后,在DOS下可用TYPE命令打印出来,不过.CRF文件一般不需要打印。

2. 链接

汇编程序产生的二进制目标文件(.OBJ文件)仍然不可执行,必须经过链接,将它转换

成.EXE文件才可执行, 链接程序为LINK.EXE, 它可以把多个模块链接在一起, 这些模块可以是库文件或汇编程序产生的目标文件。链接过程如下:

```
C: > LINK EXAMPLE
Microsoft (R) Personal Computer Linker Version 2.40
Copyright (C) Microsoft Corp 1983,1984,1985, All rights reserved
Object Modules [.OBJ]:
Run File [EXAMPLE.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning :No Stack Segment
There Was 1 Error Detected
```

在DOS下运行LINK命令后, 先显示版本号, 然后依次出现4个问题, 首先询问链接的目标文件名, 若在LINK命令后已打入目标文件名, 则此行不出现。如果链接多个目标文件, 那么可以将多个目标文件名一次输入, 中间用加号“+”连接。LINK命令要求输入的LIB文件是程序中需要用到的库文件, 若无特殊需要, 可以在遇到[.LIB]提问时直接回车。

链接命令产生两个文件, 一个为可执行的EXE文件, 对此提问用户可直接回车。另一个为MAP文件, 它是链接映像文件, 给出了每个段在存储器中的分配情况, 一般不需要此文件, 需要时打入文件名。

因为源程序没有堆栈段, 所以链接结果给出无堆栈段的警告错误, 但不影响程序执行。

如果在链接中发现错误, 那么屏幕上会显示出错信息和错误类型, 用户在修改错误后可重新进行汇编和链接。

如果汇编和链接都正确, 产生的可执行文件就可在DOS下运行, 只要直接打入文件名即可。

上述介绍的是通用的MASM 5.0版本的操作方法, 如果用MASM 6.0或更高的版本, 则汇编、链接可一步完成, 具体介绍可参见相关书籍。

2.2.2 调试程序DEBUG

在汇编及链接过程中能够检查出汇编语言源程序的语法错误, 但有些逻辑错误、结构错误只有在调试运行中才能发现。

调试工具有基于命令行方式的DEBUG和集成环境的TDEBUG和CODEVIEW等。集成环境的调试工具使用方便, 功能齐全, 界面友好, 但限于篇幅, 本书对此不作介绍。命令行方式的DEBUG的优点是便于获取, 一般操作系统中都含有该工具。以下介绍命令行的DEBUG。

DEBUG是为汇编语言设计的, 它给出了一些调试命令, 也可通过单步、断点、跟踪等方法有效地进行调试。

1. DEBUG的调用

DEBUG的命令格式都是一个字母, 后跟一个或多个参数:

字母[参数]

使用该命令的注意事项:

- 1) 字母不区分大小写。
- 2) 只使用16进制数, 没有后缀字母。
- 3) 分隔符(空格或逗号)只在两个数值之间是必须的, 命令和参数间可无分隔符。
- 4) 每个命令只有在按下回车键后才有效, 可以用Ctrl+Break中止命令的执行。
- 5) 命令如果不符合DEBUG的规则, 则将以“error”提示, 并用“^”指示错误位置。

许多命令的参数是主存逻辑地址, 形式是“段基地址: 偏移地址”。其中, 段基地址可以是段寄存器或数值; 偏移地址是数值。如果不输入段基地址, 则采用默认值, 这个默认值可以是默认的段寄存器值, 如果没有提供偏移地址, 则通常就是当前偏移地址。

在DOS提示符下, 可键入命令:

```
C: >DEBUG[D: ][Path][Filename. ext][Parm1][Parm2]
```

说明:

- D: 驱动器名
- Path: 路径名
- Filename. ext: 文件名. 扩展名
- Parm1, Parm2: 命令参数

在DEBUG命令后用户键入文件名, DEBUG将指定的文件装入内存, 若用户未键入文件名, 则DEBUG对当前存储器中的内容进行操作, 也可用N和L命令将需要的文件装入存储器。

在DEBUG程序调入后, 根据有无被调试程序及其类型相应地设置寄存器组的内容, 发出DEBUG的提示符“-”, 此时就可用DEBUG命令来调试程序。

- 运行DEBUG程序时, 如果不带被调试程序, 则所有段寄存器值相等, 都指向当前可用的主存段。除SP之外的通用寄存器都设置为0, 而SP指向这个段的尾部指示当前堆栈顶。IP=0100H, 状态标志都是清0状态。
- 运行DEBUG程序时, 如果带入的被调试程序扩展名不是.EXE, 则BX和CX包含被调试文件大小的字节数 (BX为高16位), 其他与不带被调试程序的情况相同。
- 运行DEBUG程序时, 如果带入的被调试程序扩展名是.EXE, 则需要重新定位。此时, CS: IP和SS: SP根据被调试程序确定, 分别指向代码段和堆栈段。DS: ES指向当前可用的主存段, BX和CX包含被调试文件大小的字节数 (BX为高16位), 其他通用寄存器为0, 状态标志都是清0状态。

DEBUG程序调入后, 出现提示符“-”, 此时系统已在DEBUG管理下, 可以键入DEBUG的各种命令进行调试。命令输入不区分大小写, 命令后带有参数时, 两个16进制的参数间要有分界符。利用Ctrl+Break键可中止命令, 返回DEBUG提示符。当一个命令产生多行输出时, 可用Ctrl+Lock键暂停上滚, 按任一键继续显示下面内容。

2. DEBUG的主要命令

(1) 显示存储单元命令D (Dump)

该命令有以下几种格式:

- -D 地址: 从指定地址起显示80个字节的内容。