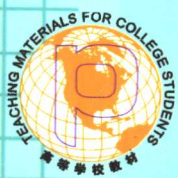
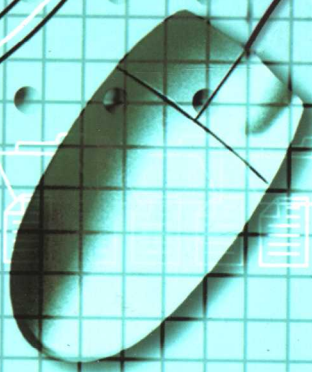
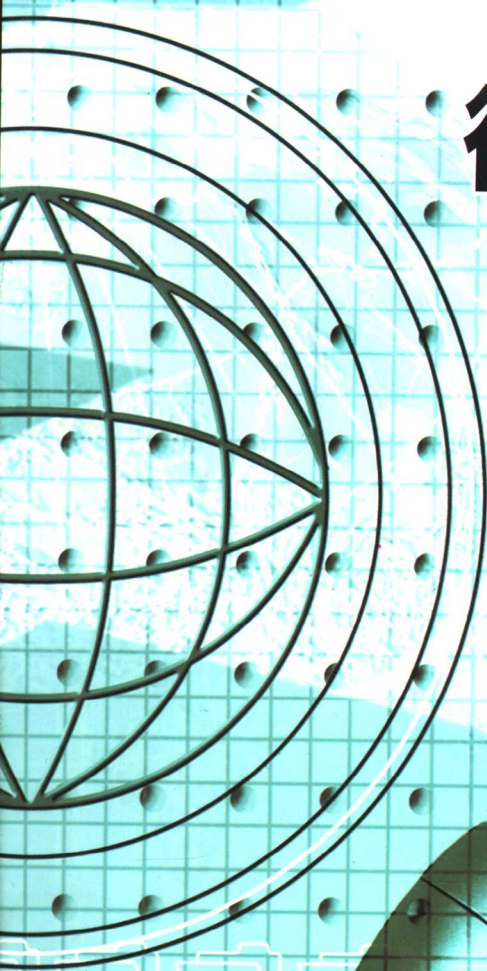


◎ 戴永寿 周海滨 雷国江 主编

微机原理汇编 及接口设计



中国石油大学出版社



TEACHING MATERIALS FOR COLLEGE STUDENTS

高等学校教材

内容简介

微机原理汇编及接口设计

戴永寿 周海滨 雷国江 主编

中国石油大学出版社

图书在版编目(CIP)数据

微机原理汇编及接口设计/戴永寿等主编. —东营:
中国石油大学出版社, 2006. 8
ISBN 7-5636-2196-2

I. 微... II. 戴... III. ①微型计算机—理论—高等
学校—教材②微型计算机—接口—高等学校—教材
IV. TP36

中国版本图书馆 CIP 数据核字(2006)第 093762 号

中国石油大学(华东)规划教材
书 名: 微机原理汇编及接口设计
作 者: 戴永寿 周海滨 雷国江

责任编辑: 袁超红(电话 0546—8395745)
封面设计: TRANSMART

出 版 者: 中国石油大学出版社(山东 东营 邮编 257061)
网 址: <http://www.uppbook.com.cn>
电子信箱: shiyoujiaoyu@126.com
排 版 者: 海讯科技有限公司
印 刷 者: 石油大学印刷厂
发 行 者: 中国石油大学出版社(电话 0546—8391797, 8392791)
开 本: 180×235 印张: 7.875 字数: 158 千字
版 次: 2006 年 8 月第 1 版第 1 次印刷
定 价: 11.00 元

内 容 简 介

本书是微机原理及其接口技术理论课程的配套实验教材。教材按照多层次、模块化实验体系进行编写,包括汇编语言程序设计和接口技术应用两部分。每部分各有8个实验模块,共计16个实验模块。每个实验模块又包括基本实验项目、选做或开放实验项目。针对不同专业 and 不同学时要求,书中设计了多种模块化实验内容选择方式,并突出了设计性和综合性实验内容的训练,同时为自主开放性实验提供了系统的实践训练内容。

本书可用作自动化、电子信息、电气工程、测控技术、应用物理及机电类相关专业的实验教材,也可作为微机原理及其接口技术理论课程的辅助学习教材。

P r e f a c e

前 言

计算机技术的飞速发展为工业测控领域提供着日新月异的新方法、新技术和新的应用系统,它们已逐步渗透融合进工业自动化、电气信息、测控技术与仪器、机电一体化等工程技术领域。今天,无论是在日常生活中,还是在各行各业的工作中,人们已经清楚地认识到,处于核心地位的计算机技术正发挥着越来越重要的作用。

众所周知,微机原理及其接口技术是计算机应用技术的基础,尤其在工业测量与控制领域。尽管计算机测控系统形式多样、功能不一,但是,它们必然都是由计算机或微机和各种软硬件接口所构成的一个有机系统。这样的系统不仅要求能完成特定的功能,还必须能够根据测控对象的特性进行系统功能的扩展和更新。因而,掌握微机工作原理和接口设计软硬件开发的基本知识和必备技能,才能根据测控对象的特点和功能要求,设计集成,扩充完善计算机测控系统在工程领域中的应用。

“微机原理”是高等学校电气信息类专业的一门实践性很强的技术基础课程。本书作为微机原理及其接口设计理论课程的配套实验教材,在强调基本实验技能训练、加强学生对微机基本概念和基本原理理解的基础上,突出了设计性和综合性实验内容的训练。同时为自主开放性实验提供了系统的实践训练内容,注重加强学生分析问题、解决问题的能力,并注重学生综合应用能力和自主创新意识的培养。

按照精品课程建设标准和人才培养目标要求,本教材以多层次、模块化实验体系进行编写。针对不同专业和不同学时要求,设计了多种模块化实验内容选择方式,可按照实际情况选择实验的种类和实验项目个数,实验选择方式灵活。针对不同专业的知识点需求,从实验内容上提供多种组合方式,保证实验训练更具针对性和专业特点。实验内容从浅入深,符合教学大纲的基本规律。另外,本实验教材还设置了选做实验内容,供任课教师参考采用,也适用于学生自主开放性实验。

本书是“微机原理”课程组长期教学实践工作经验积累的成果。参加本书编写的有戴永寿、周海滨、雷国江、孙洪涛。此外,白蓉、邬颜辉、吴荔清、李立刚、杨华、谢一飞、唐永红和陈真等老师对本书的编写提出了许多宝贵意见,并给予了大力支持和帮助,在此一并表示衷心的感谢。

由于编者水平有限,书中错误和不妥之处在所难免,敬请读者批评指正。

编 者

2006年7月

G u i d e

使用说明

各实验项目中的标记说明和基本要求：

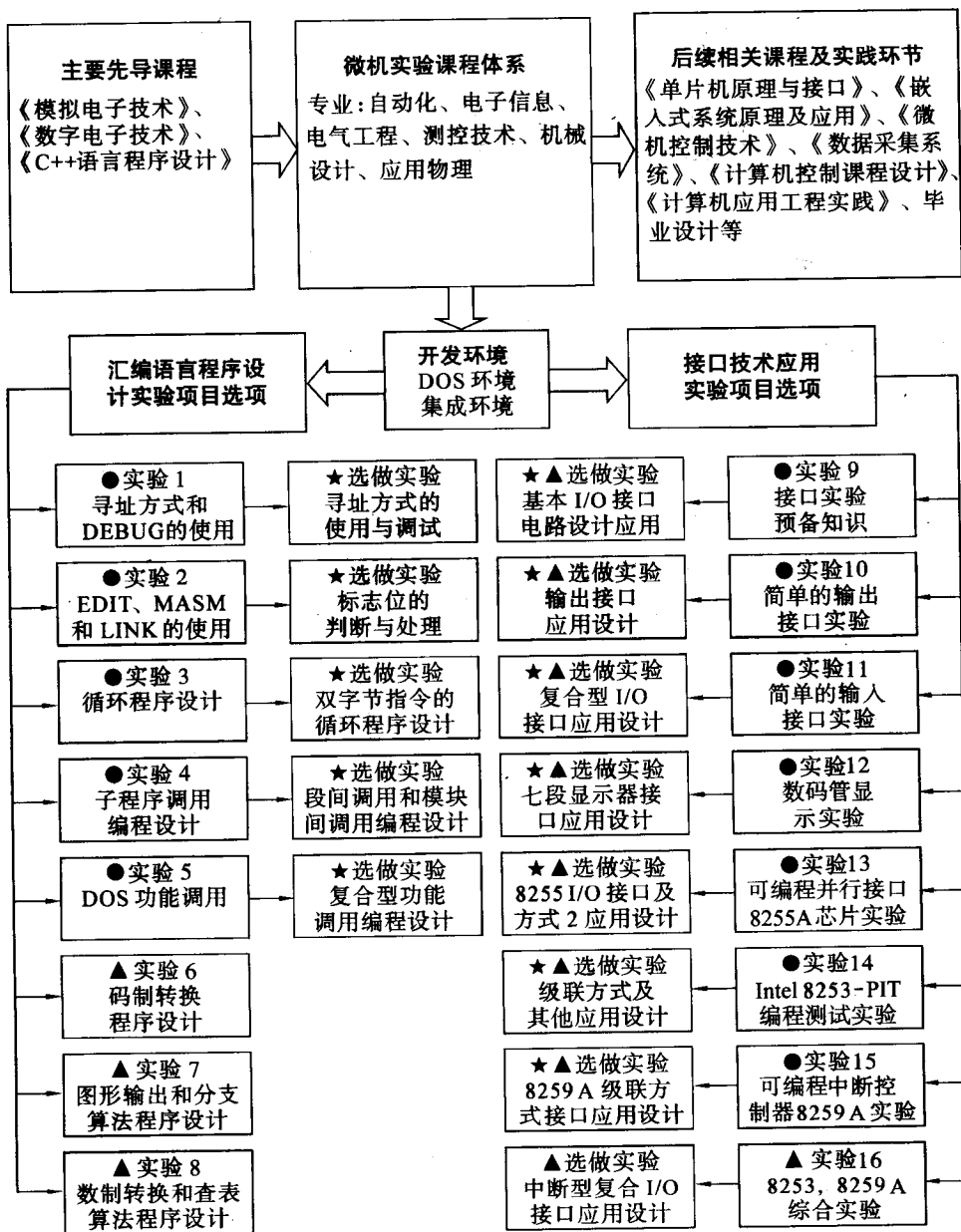
● **基本实验项目：**自动化、电子信息、电气工程、测控技术、应用物理等专业（32学时）必做实验项目，属于验证性和基本应用设计性项目。对于基本实验项目，要求学生根据实验条件、预习要求和实验内容要求的步骤，在规定的实验课时内完成。

★ **选做实验项目：**主要是为信息类专业提供的提高应用设计能力的选做实验项目，也是设计性开放实验的选做项目。对于选做的设计性实验项目，要求学生根据实验条件和实验内容要求，在基本实验项目完成后的时间内灵活选择，自行完成实验方案的规划、实验步骤的设计和实验结果的分析。

▲ **开放实验项目：**主要是电子信息、测控技术和自动化等专业提高综合应用能力的设计性开放实验选做项目。对于开放的设计性实验项目，要求学生根据实验条件和实验内容要求，自行完成实验方案的规划、实验步骤的设计、实验过程的进行和实验结果的分析。实验项目在实验室规定的开放时间里完成，教师提供适当的指导。设计性开放实验项目注重的是学生创新能力、独立动手能力和综合实践能力的训练。

实验预习要求：在实验前一定要认真预习与每次实验有关的内容和知识。根据每次实验任务的要求，编写出源程序并画出接口电路原理图。实验课开始前，实验指导老师应认真检查每一位学生的实验预习报告。

实验总学时少于32学时的其他专业，按照课程实际教学内容和学时数，在基本实验项目中选择安排实验项目。下图为本教材实验体系的模块化结构和实验项目的选择指导图。



本教材实验体系的模块化结构和实验项目指导图

目 录

CONTENTS

第1部分 汇编语言程序设计

第1章 汇编语言程序设计过程	3
● 1.1 汇编语言程序设计过程概述	3
1.1.1 源程序的编辑	3
1.1.2 源程序的汇编	3
1.1.3 链接	4
1.1.4 程序调试	4
● 1.2 汇编程序开发环境概述	4
1.2.1 Windows 集成环境	4
1.2.2 DOS 环境	5
第2章 调试工具 DEBUG	9
● 2.1 DEBUG 程序的调用	9
● 2.2 DEBUG 命令的基本使用方法	10
2.2.1 显示存储单元命令 D (Dump)	10
2.2.2 修改存储单元内容命令 E (Enter)	11
2.2.3 检查和修改寄存器内容命令 R (Register)	12
2.2.4 汇编命令 A (Assemble)	13
2.2.5 反汇编命令 U (Unassemble)	14
2.2.6 运行命令 G (Go)	16
2.2.7 跟踪命令 T (Trace)	17
2.2.8 退出命令 Q (Quit)	17
2.2.9 其他命令	17

第3章 EDIT 编辑器	20
● 3.1 DOS系统EDIT的基本使用方法	20
3.1.1 屏幕显示	20
3.1.2 功能键	22
3.1.3 使用File下拉菜单	22
3.1.4 修改已存在的文本	24
3.1.5 使用Edit下拉菜单	25
3.1.6 使用Search搜索下拉菜单	25
● 3.2 用EDIT建立源文件的推荐格式	25
● 3.3 Windows下编辑源文件的方法	26
第4章 汇编程序MASM的使用方法	28
● 4.1 用MASM汇编源文件的一般格式	28
● 4.2 用MASM汇编源文件的推荐格式	29
第5章 链接程序LINK的使用方法	30
● 5.1 用LINK链接程序的一般格式	30
● 5.2 用LINK链接程序的推荐格式	31
第6章 集成开发环境使用方法	32
● 6.1 编辑源文件	32
● 6.2 汇编	33
● 6.3 链接	34
● 6.4 程序运行	34
第7章 BIOS中断和DOS功能调用	36
● 7.1 BIOS中断和DOS功能调用概述	36
● 7.2 DOS系统功能调用方法	38
7.2.1 DOS键盘功能调用	39
7.2.2 DOS显示功能调用	41
7.2.3 程序结束系统功能调用	43
第8章 汇编语言程序设计实验	44
● 实验1 寻址方式和DEBUG的使用	44

● 实验2 EDIT, MASM 和 LINK 的使用	47
● 实验3 循环程序设计	48
● 实验4 子程序调用程序设计	49
● 实验5 DOS 功能调用	50
● 实验6 ASCII 码转换成二进制数的程序设计	51
● 实验7 综合程序设计(1)	52
● 实验8 综合程序设计(2)	53

第2部分 接口技术应用

第9章 微机接口实验系统	57
● 9.1 微机接口实验装置介绍	57
9.1.1 实验装置的组成	57
9.1.2 接口板部分电路介绍	57
9.1.3 接口板的安装	61
● 9.2 接口实验技术指导	63
9.2.1 输出接口电路调试	63
9.2.2 输出接口软件调试	63
9.2.3 输入接口电路调试	65
● 9.3 微机接口实验装置的安全性	66
第10章 接口技术应用实验	67
● 实验9 接口实验预备知识	67
● 实验10 简单的输出接口实验	72
● 实验11 简单的输入接口实验	75
● 实验12 数码管显示实验	78
● 实验13 可编程并行接口 8255A 芯片实验	81
● 实验14 Intel 8253-PIT 编程测试实验	86
● 实验15 可编程中断控制器 8259A 实验	91
● 实验16 8253, 8259A 综合实验	102
附录1 汇编程序出错信息	103
附录2 常用的 54/74 系列集成电路芯片	111
参考文献	114

第 1 部分

Part One

■ 汇编语言程序设计 ■

第 1 章 汇编语言程序设计过程

本章介绍汇编语言程序设计的基本步骤,主要目的是了解汇编程序的开发过程,各步骤的详细介绍请参考有关章节。

1.1 汇编语言程序设计过程概述

在学习了高级语言如 C/C++ 程序设计后,用户对程序的开发过程已有了解,并建立了清晰的程序设计步骤,即编辑、编译、链接、运行和调试。同样,在进行汇编语言程序设计和开发时,也需要类似的步骤。图 1.1 给出了汇编语言程序设计的基本步骤。

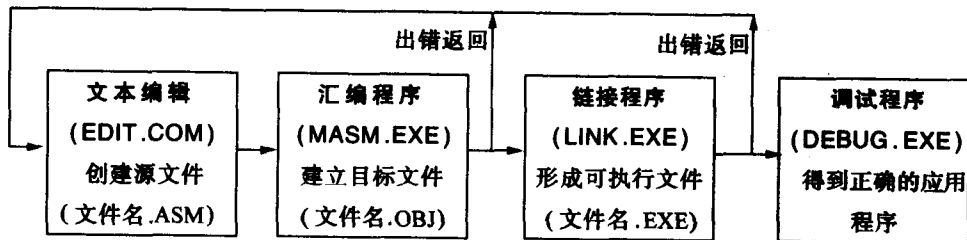


图 1.1 汇编语言程序设计的过程

1.1.1 源程序的编辑

用户首先要用文本编辑器产生自己的汇编语言源程序(扩展名为 ASM 的文件),而汇编语言源程序就是用语句(MASM 中可使用伪指令语句和指令性语句)编写的一种文本文件。编辑文本文件的工具较多,如 EDIT、Turbo C 2.0、Windows 记事本和 MASM 集成环境等。具体选择用哪一种,按用户的习惯决定。

1.1.2 源程序的汇编

汇编语言源程序是不能为计算机直接识别的,所以用户要用汇编程序(如 MASM.EXE)把源程序翻译成由机器代码构成的浮动目标文件(扩展名为 OBJ 的文件)。目标文件虽然已经是二进制目标文件,但它还不能直接执行。

1.1.3 链接

用户程序要想成为可执行程序,必须要经过链接程序(LINK. EXE)把目标文件与库文件合成一个可执行程序(扩展名为 EXE 或 COM 的文件)。这个文件在 DOS 操作系统下只要输入文件名就可以执行。

1.1.4 程序调试

链接成功后的可执行文件虽说可以运行,但还不一定是用户所需的最终应用程序。当程序运行后出现诸如结果不对或是结果输出格式不妥等非语法错误时,这就需要用户重新对自己的程序进行更加细致的分析和检查,找出原因并解决它。有时,程序员还想进一步了解程序执行过程中的其他细节,如 CPU 寄存器内容的变化过程、堆栈变化过程、内存内容的变化过程等,这就需要借助调试工具来完成,也就是图 1.1 中程序设计过程的最后一步。汇编语言程序设计过程中常用的调试工具之一就是 DEBUG。

以上过程中,当汇编或链接后出现错误提示时,程序员需要返回到最初的步骤(即编辑状态下)对源文件进行检查和修改,然后再逐步进行。

总结上述内容,汇编程序的开发过程可分为以下步骤:

- (1) 用文本编辑器创建源文件(文件名. ASM);
- (2) 用汇编程序建立目标文件(文件名. OBJ);
- (3) 用链接程序形成可执行文件(文件名. EXE);
- (4) 在操作系统下直接执行或用 DEBUG 调试可执行文件。

1.2 汇编程序开发环境概述

前一节我们已经简单地叙述了汇编语言程序从建立到执行的过程。要完成这一过程,在计算机系统中就要有相应的环境或程序。DOS 环境下一般要有编辑程序、汇编程序、链接程序和调试程序。基于 Windows 的开发环境多为集成开发工具,如 Visual ASM, Wen MASM 等。在实际开发应用中,程序员可按照各自的条件灵活选择开发环境。下面将简单介绍这两类开发环境的基本应用方法。

1.2.1 Windows 集成环境

用于汇编语言程序开发的集成环境种类较多,其共有的基本特点是:开发过程的各步骤功能都基于同一平台下。图 1.2 所示为 Visual ASM 的界面示例。

图 1.2 是已打开源文件(lei11. asm)的 Visual ASM 界面示例。可见, Visual ASM 与 Windows 环境下的其他语言开发平台风格相似。对于使用过 Windows

环境下其他语言开发平台(如 Visual C++ 6.0)的用户来说,其使用方法易于掌握。

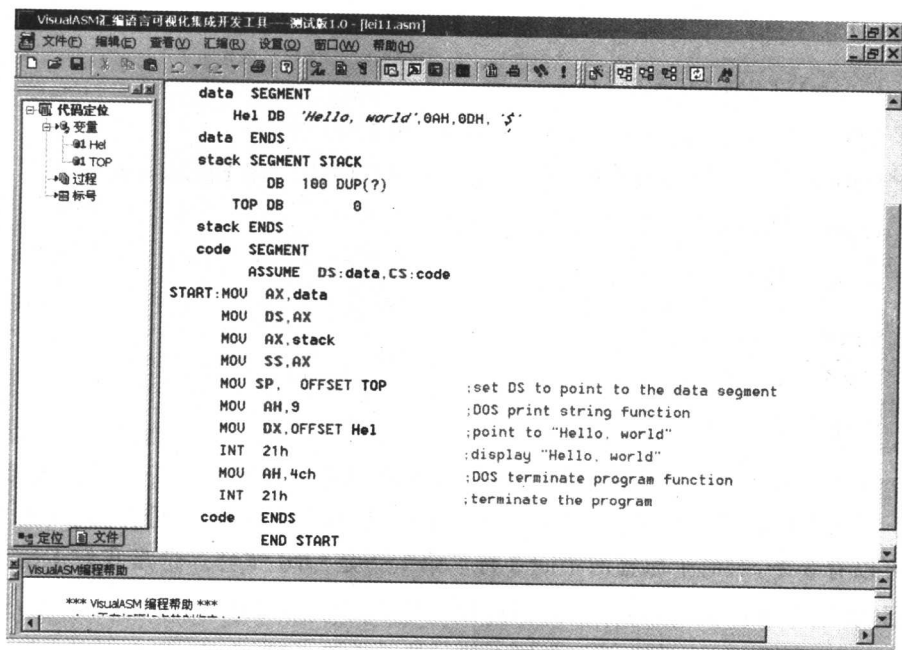


图 1.2 Visual ASM 界面

1.2.2 DOS 环境

在 DOS 环境下,要完成汇编程序的开发过程,一般要有如下文件:

- (1) 文本编辑器;
- (2) MASM. EXE;
- (3) LINK. EXE;
- (4) DEBUG. EXE。

下面以使用编辑程序 EDIT 为例,概述汇编程序开发的基本步骤。

1) 创建 ASM 文件

若用 EDIT 创建源程序,当用户源文件名为 MOVES 时,其 DOS 下的命令格式为:

```
C:\>EDIT MOVES. ASM ↵
```

命令末尾的符号“↵”代表按下 1 次回车键(Enter 键)。

执行上述命令,就可以在 C 盘根目录下创建以 MOVES. ASM 为文件名的源文件。

用 EDIT 创建源文件的详细使用方法请参考第 3 章。

2) 用 MASM 程序产生 OBJ 文件

源文件创建后,就要用汇编程序对源文件进行汇编。汇编后产生二进制目标文件(OBJ 文件)。其操作与汇编程序回答如下:

```
C:\>MASM MOVES ↵
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981—1985,1987. All rights reserved.
Object filename [MOVES.OBJ]:↵
Source listing [NUL.LST]:↵
Cross-reference [NUL.CRF]:↵
0 Warning Errors
0 Severs Errors
```

这样汇编正确后产生的目标文件(MOVES.OBJ)就保存在 DOS 的当前路径下。汇编程序的输入文件是 ASM 文件。按以上 DOS 命令格式进行时,其输出文件可以有 3 个,对应于回答的中间 3 行。第 1 个是 OBJ 文件。这是我们汇编的主要目的,这个文件我们需要的。对[MOVES.OBJ]后的回答应按“↵”键,这样就在 C 盘根目录下建立了这一目标文件。第 2 个是 LST 文件,称为列表文件。这个文件同时列出源程序和机器语言程序清单,并给出符号表,因而可使程序调试更加方便。这个文件是可有可无的。如果不需要,则对[NUL.LST]后的回答按“↵”键;如果需要这个文件,则可回答文件名,本例中可以是 MOVES。这样 MOVES.LST 就建立起来了。汇编程序能提供的第 3 个文件是 CRF 文件,这个文件用来产生交叉引用表 REF。一般程序不需要建立此文件,所以可以对[NUL.CRF]用“↵”来回答,这样就完成了汇编过程。

用户若要用到 LST 文件和 CRF 文件,请参考汇编程序的相关资料。

3) 用 LINK 程序产生 EXE 文件

汇编程序已产生出二进制目标文件(OBJ 文件),但 OBJ 文件并不是可执行的文件,还必须使用链接程序(LINK)把 OBJ 文件转换为可执行的 EXE 文件。当一个程序是由多个模块组成时,也应该通过 LINK 把它们链接在一起。LINK 程序操作方法及程序回答如下:

```
C:\>LINK MOVES ↵
Microsoft (R) Overlay Linker Version 3.60
Microsoft (C) Microsoft Corp 1983—1987. All rights reserved.
Run File[MOVES.EXE]:↵
```

```
List File[NUL. MAP];MOVES ↵
Libraries[. LIB];↵
LINK:warning L4021:no stack segment
There was 1 error detected.
```

LINK 程序有两个输入文件 OBJ 和 LIB。OBJ 是我们需要链接的目标文件, LIB 则是程序中需要用到的库文件。如无特殊需要,则应对 [LIB] 回答“↵”。LINK 程序有两个输出文件。一个是后缀为 EXE 的文件,这当然是我们需要的,应对 [MOVES. EXE] 回答“↵”,这样就在 DOS 的当前路径下建立了该可执行文件。LINK 的另一个输出文件为 MAP 文件,它是链接程序的列表文件,又称为链接映象(Link Map),它给出每个段在存储器中的分配情况。下面所示的内容是 MOVES. MAP 文件的示例。

```
Warning: No STACK segment
Start      Stop      Length    Name      Class
0000H      001DH     001EH     CODE
0020H      0047H     0028H     DATA
0050H      0077H     0028H     EXTRA
Origin Group
Program entry point at 0000:0000
```

链接程序给出的无堆栈段的警告性错误并不影响程序的执行,所以,到此为止,链接过程已经结束,用户可以执行 MOVES 程序了。

4) 程序的执行和调试

在建立了 EXE 文件后,就可以直接在 DOS 下执行程序,如下所示(假设 MOVES. EXE 就在 C 盘当前路径下):

```
C:\>MOVES ↵
C:\>
```

程序运行结束并返回 DOS。如果用户要想在计算机的显示器屏幕上显示相关的运行结果,那么程序中必须要有相关显示输出的指令,否则显示器屏幕上不会有运行结果。

此外,大部分程序必须经过调试阶段才能纠正程序执行中的错误,特别是算法错误。那么又怎样来调试程序呢? 这就要使用 DEBUG 程序。这里我们只介绍 DEBUG 的进入。若要对前面已建立的程序 MOVES. EXE 进行调试,就要先把它装入 DEBUG,操作如下: