

● 电工电子实验教学规划示范教材



微机原理与 接口实验

◎ 董毅 林卫 程世平

华中科技大学出版社
<http://www.hustp.com>

电工电子实验教学规划示范教材

微机原理与接口实验

董毅 林卫 程世平

华中科技大学出版社

中国·武汉

图书在版编目(CIP)数据

微机原理与接口实验/董毅 林卫 程世平. —武汉:华中科技大学出版社,
2008年9月

ISBN 978-7-5609-4774-7

I. 微… II. ①董… ②林… ③程… 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(2008)第 113856 号

微机原理与接口实验

董毅 林卫 程世平

策划编辑:张志华

责任编辑:姚幸

责任校对:汪世红

封面设计:刘卉

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录排:华中科技大学惠友文印中心

印刷:湖北恒泰印务有限公司

开本:787mm×960mm 1/16

印张:15.5

字数:280 000

版次:2008年9月第1版

印次:2008年9月第1次印刷

定价:25.00元

ISBN 978-7-5609-4774-7/TP·660

(本书若有印装质量问题,请向出版社发行部调换)

内 容 提 要

本书以 80x86 系列微机为实验对象,实验内容按照微机原理实验课程教学大纲来编排,覆盖了汇编语言程序设计和硬件系统设计的全部内容。

本书重视基础,体现创新,可作为高等学校工科类各专业本科和专科学生的实验教材,也可作为从事计算机系统应用开发的工程技术人员的设计参考书。

前 言

微机原理与接口技术是高等学校工科各专业的专业基础课程之一,不但要求学生要有较扎实的专业基础知识,而且还要求有实际的动手能力。本书是为配合微机原理与接口技术理论课教学而编写的,目的在于帮助学生深入理解微机原理和接口的专业知识,增强分析和解决问题的能力,提高设计和调试微机应用系统的能力。

本书所选用的实验内容是在华中科技大学通信与信息系统专业、电子与通信工程专业、自动控制专业、生物医学专业、软件工程专业以及华中科技大学提高班等班级的微机原理与接口技术实验教学实践基础上提炼出来的,本书既重视基础,又体现创新。虽然现在利用实验装置进行微机原理与接口技术实验已经成为一种趋势,但是我们仍然强调学生应当在实验装置的基础上进行一定的扩展,利用面包板(插接板)和外接电路完成实验。本书中很多实验都体现了这样的思想,如基本 I/O 实验,ADC0808/0809 实验等。

本书按照先软件后硬件的顺序安排实验,这与理论课程的教学顺序是一致的。这样可以保证实验课和理论课同时进行,方便教学计划的编排。本书第 1 章为汇编语言开发入门,介绍汇编语言开发的基本过程,使学生能够迅速地入门。第 2 章为汇编语言开发环境,详细介绍汇编语言开发过程中所使用的软件环境和使用方法。第 3 章为汇编语言程序设计实验,介绍汇编语言程序设计的基础方法与技巧。第 4 章为硬件实验环境,介绍进行硬件实验的环境及其检测方法。第 5 章为硬件接口实验,介绍常用的硬件接口设计和实现方法。

本书由董毅、林卫、程世平负责编写,由董毅负责统稿。参加本书编写和实验验证工作的人员还有卢钢、张士军、杨明、左冬红、王红等。

崔厚孝、陈任等老师为本书的编写给予了大力的支持和帮助,提出了很多宝贵的意见;华中科技大学出版社对本书的出版给予了全力支持,付出了辛勤的劳动,在此一并表示衷心的感谢。

由于编者水平有限,时间仓促,书中不妥之处在所难免,敬请广大读者批评指正。

编 者

于华中科技大学

2008 年 8 月

目 录

第 1 章 汇编语言开发入门	(1)
1.1 DOS 环境下汇编快速入门	(1)
1.2 Windows 环境下汇编快速入门	(3)
第 2 章 汇编语言开发环境	(6)
2.1 汇编程序结构	(6)
2.2 汇编过程	(7)
2.3 用 EDIT 建立源程序文件	(8)
2.4 宏汇编程序 MASM	(10)
2.5 链接程序 LINK	(12)
2.6 汇编链接程序 ML	(14)
2.7 EXE 文件的运行	(15)
2.8 DEBUG 调试	(15)
2.9 DOS 可执行程序结构和内存映像图	(23)
2.10 TASM/TLINK/TD 介绍	(27)
2.11 DOS 操作系统简介	(31)
第 3 章 汇编语言程序设计实验	(35)
3.1 实验及实验报告要求	(35)
3.2 DOS 及 BIOS 功能调用	(36)
3.3 顺序程序结构	(37)
3.4 分支程序结构	(40)
3.5 循环程序结构	(44)
3.6 子程序设计	(51)
3.7 C/C++ 混合语言编程	(61)
第 4 章 硬件实验环境	(62)
4.1 微机接口实验装置简介	(62)
4.2 实验装置的检测	(65)
4.3 硬件接口设计过程	(68)
第 5 章 硬件接口实验	(70)
5.1 实验及实验报告要求	(70)
5.2 基本输入接口(1)	(70)

5.3	基本输入接口(2)	(74)
5.4	基本输出接口(1)	(78)
5.5	基本输出接口(2)	(82)
5.6	基本输入/输出混合接口(1)	(85)
5.7	基本输入/输出混合接口(2)	(89)
5.8	存储器 6264 的 I/O 实验	(93)
5.9	存储器 6116 的 I/O 实验	(97)
5.10	硬件中断实验	(102)
5.11	PCI 中断实验	(110)
5.12	模/数转换器 ADC0808/0809	(116)
5.13	模/数转换器 AD574	(126)
5.14	数/模转换器 DAC0832	(135)
5.15	并行接口芯片 8255	(140)
5.16	七段 LED 显示器接口	(156)
5.17	4×2 矩阵式键盘接口	(160)
5.18	4×4 矩阵式键盘接口	(164)
5.19	4×2 键盘和七段 LED 显示器的联合接口	(169)
5.20	4×4 键盘和七段 LED 显示器的联合接口	(174)
5.21	可编程计数器/定时器 8253/8254	(179)
附录 A	TD-PIT 实验装置简介	(195)
附录 B	TD-PIT+ 实验装置简介	(203)
附录 C	8086 指令系统表	(213)
附录 D	8086 算术逻辑运算指令对状态标志位的影响	(218)
附录 E	ASCII 码字符表	(219)
附录 F	MASM 汇编程序出错信息表	(221)
附录 G	Turbo Debugger 主菜单命令	(228)
附录 H	Turbo Debugger 局部菜单命令	(231)
附录 I	本书所用部分集成电路引脚图	(236)
参考文献		(240)

第 1 章 汇编语言开发入门

在这一章里,通过实例,给出 DOS 和 Windows 环境下编写汇编语言程序和汇编的全过程,使学生能够迅速熟悉汇编语言软件开发环境。

1.1 DOS 环境下汇编快速入门

启动计算机,进入 DOS 操作系统,显示提示符

C:\>

按以下步骤完成汇编语言程序 HELLO. ASM 的汇编过程,运行该程序后在屏幕上显示字符“Hello World!”。

步骤 1 进入工作目录。

键入 D:并回车,即

C:\>D: (↵表示按“回车”键,下同)

D:\>

步骤 2 运行 EDIT 文本编辑器,编辑 HELLO. ASM 源文件。

键入 EDIT HELLO. ASM 并回车,即

D:\>EDIT HELLO. ASM ↵

EDIT 编辑界面如图 1.1 所示。

步骤 3 在 EDIT 界面编辑以下源程序。

```
stack segment stack
    dw 32 dup (?)
stack ends
data segment
    OBUF DB 'Hello World! $'
data ends
code segment
    assume ss:stack, ds:data, cs:code
begin proc far
    push ds
    sub ax,ax
    push ax
```

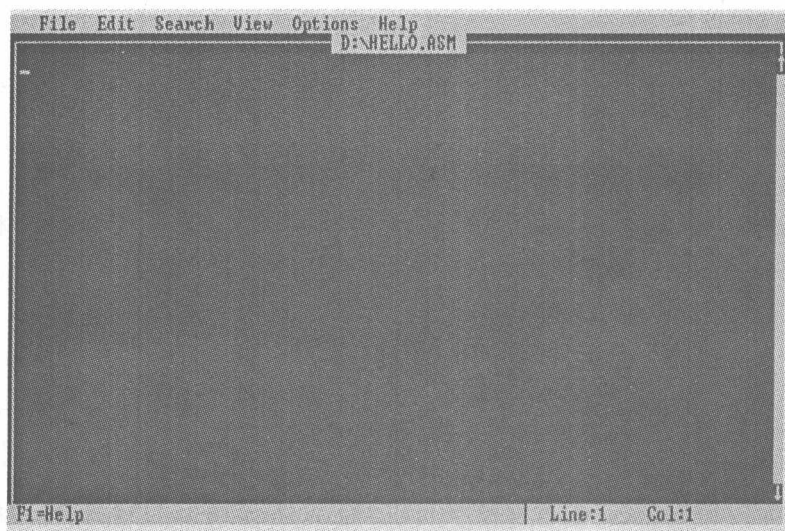



图 1.1 EDIT 界面

```

mov ax,data
mov ds,ax
MOV DX,OFFSET OBUF
MOV AH,9
INT 21H
ret
begin endp
code ends
end begin

```

步骤 4 保存程序并退出。

按 ALT-F (“ALT-F”表示“ALT”键和“F”键同时按下,下同)进入 File 菜单,选择 Save;再次按 ALT-F 进入 File 菜单,选择 Exit,退出 EDIT 文本编辑器。

步骤 5 汇编 HELLO.ASM 生成目标文件 HELLO.OBJ。

键入 MASM HELLO.ASM 并回车,即

D:\>MASM HELLO.ASM ↵

屏幕上显示

Microsoft (R) MASM Compatibility Driver

Copyright (C) Microsoft Corp 1993. All rights reserved.

Invoking: ML.EXE /I. /Zm /c /Ta HELLO.ASM

Microsoft (R) Macro Assembler Version 6.11

Copyright (C) Microsoft Corp 1981—1993. All rights reserved.

Assembling: HELLO.ASM

屏幕没有错误提示信息,表示汇编通过,这时在当前目录下会生成 HELLO.OBJ 文件。如果屏幕显示错误或者警告信息,则说明源文件编辑有问题,返回步骤 2,检查修改文件,直至没有错误为止。

步骤 6 链接 HELLO.OBJ 生成可执行文件。

键入 LINK HELLO.OBJ;并回车,即

D:\>LINK HELLO.OBJ;↵

屏幕上显示

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992

Copyright (C) Microsoft Corp 1984—1992. All rights reserved.

屏幕没有错误提示信息,表示链接通过,这时在当前目录下会生成 HELLO.EXE 文件。如果屏幕显示错误或者警告信息,则说明源文件编辑有问题,返回步骤 2,检查修改文件。直至没有错误为止。

步骤 7 运行生成可执行文件。

键入 HELLO 并回车,即

D:\>HELLO ↵

屏幕上显示

Hello World!

1.2 Windows 环境下汇编快速入门

启动计算机,进入 Windows 操作系统。

按以下步骤完成汇编语言程序 HELLO.ASM 的汇编过程,该程序输出字符“Hello World!”。

步骤 1 运行“未来汇编”程序。

“未来汇编”程序是 Windows 环境下的一个国产集成编辑器,通过封装 TASM 汇编程序,给习惯使用 Windows 的用户一个窗口工作环境。“未来汇编”运行界面如图 1.2 所示。

步骤 2 在“未来汇编”编辑器中输入如下程序,文件名为 HELLO.ASM。

```
stack segment stack
```

```
dw 32 dup (?)
```

```
stack ends
```

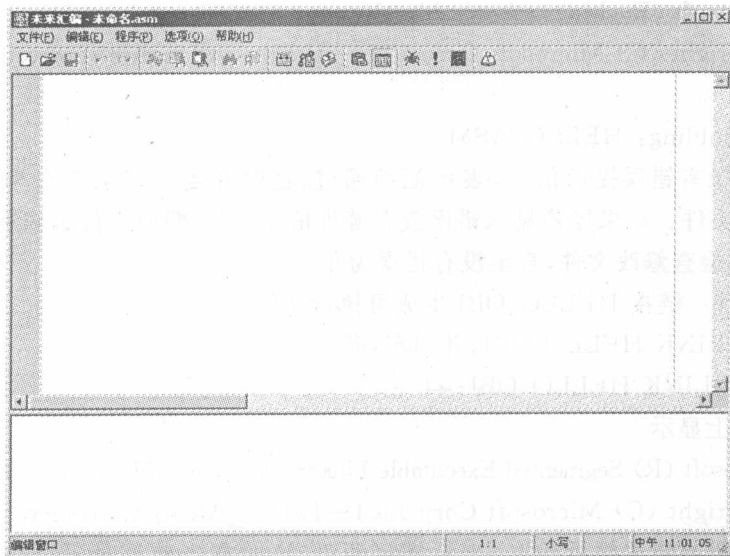


图 1.2 “未来汇编”运行界面

```

data    segment
        OBUF DB 'Hello World! $'
data    ends
code    segment
        assume ss:stack, ds:data, cs:code
begin   proc far
        push ds
        sub ax,ax
        push ax
        mov ax,data
        mov ds,ax
        MOV DX,OFFSET OBUF
        MOV AH,9
        INT 21H
        ret
begin   endp
code    ends
        end begin

```

步骤 3 汇编 HELLO.ASM,生成目标文件 HELLO.OBJ。

点击工具栏中“编译”按钮,汇编源程序,如图 1.3 所示。



图 1.3 “未来汇编”的“编译”按钮

编译过程中的提示信息显示在提示窗口中,如果“Error messages”和“Warning Messages”均为“None”,说明汇编通过,如图 1.4 所示。否则说明源文件有问题,需要修改源程序然后重新汇编,直至通过为止。

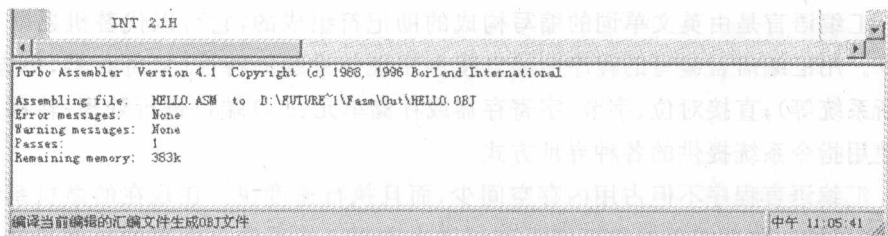


图 1.4 “未来汇编”提示信息窗口

步骤 4 链接目标文件 HELLO.OBJ。

点击工具栏“链接”按钮链接目标文件,如图 1.5 所示。



图 1.5 “未来汇编”的“链接”按钮

步骤 5 运行可执行文件 HELLO.EXE。

点击工具栏“运行”按钮,运行 HELLO.EXE,如图 1.6 所示。

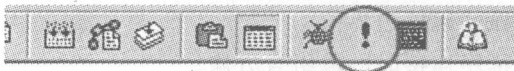


图 1.6 “未来汇编”的“运行”按钮

屏幕显示一个 DOS 窗口,内容为

未来汇编 中文学习版 1.00

Hello World!

press any key to continue ...

按任意键,返回“未来汇编”编辑界面。

第 2 章 汇编语言开发环境

2.1 汇编程序结构

汇编语言是由英文单词的缩写构成的助记符组成的,它可以代替机器语言指令。用汇编语言编写的程序能够直接利用硬件系统的特性(如寄存器、标志、中断系统等);直接对位、字节、字寄存器或存储单元、I/O 端口进行处理;转移直接使用指令系统提供的各种寻址方式。

汇编语言程序不但占用内存空间少,而且执行速度快。在现在的微机系统中,底层的一些功能仍然靠汇编语言程序来实现。汇编语言程序的效率通常高于高级语言程序的效率。

一个完整的汇编语言程序包含若干个段。段定义伪指令是表示一个段开始和结束的命令。80x86 的汇编语言完整段定义伪指令的格式为

```
段名 SEGMENT [定位方式] [组合方式] ['类型名']  
:  
段名 ENDS
```

段名由用户指定。对于数据段、附加段和堆栈段来说,段内一般是存储单元的定义、分配等伪指令语句。

代码段主要是指令及伪指令语句。

定位方式说明段的起始边界值。

组合方式说明程序连接时的段组合方法。

类型名在单引号中给出连接时组成段组的类型名。连接程序可把相同类别的段的位置靠在一起。

段和段寄存器的关系使用 ASSUME 伪指令建立,说明哪个段是代码段,哪个段是数据段。其格式为

```
ASSUME 段寄存器名:段名,...
```

段寄存器名必须是 CS、DS、ES 和 SS 中的一个,而段名必须是由 SEGMENT 定义的段名。

本书采用完整段定义的汇编语言指令,使用的典型的汇编语言程序结构如下。

错 ;定义堆栈段及堆栈长度,即使没有使用也应当定义堆栈段,否则汇编会报错

```
stack segment stack
    dw 32 dup (?)
stack ends
;定义数据段
data segment
    :
data ends
;定义代码段
code segment
    assume ss:stack, ds:data, cs:code
begin proc far ;将汇编源程序视为 DOS 下的子程序
    push ds ;将程序段前缀 PSP 进栈
    sub ax,ax
    push ax
    mov ax,data ;设置用户数据段
    mov ds,ax
    :
;返回到 PSP,执行 PSP 中的第一条指令 INT 20H,结束程序返回调用程序
    ret
begin endp
code ends
end begin
```

2.2 汇编过程

在 80x86 系列微机上运行汇编语言源程序的步骤如下。

步骤 1 生成源程序文件。

用文本编辑软件,如 EDIT、Ultraedit 等,建立或修改汇编语言源程序。汇编语言程序的扩展名必须为 .ASM。

步骤 2 汇编。

用宏汇编程序,如 MASM、TASM 等,对汇编语言源程序文件进行汇编,生成扩展名为 .OBJ 的目标文件。由于汇编所得到的目的代码的存放地址并不是可执行的绝对地址,而是浮动的相对地址。因此,必须进行链接之后才能调试和

运行。

步骤3 链接。

用链接程序,如 LINK、TLINK 等,链接目标文件和库文件,生成可执行文件。可执行文件的后缀名一般为 .EXE。能够生成可执行文件只说明汇编语言程序的编写符合语法规则,并不一定表示该程序没有其他问题。比如算法问题、逻辑问题等只有在执行阶段才能表现出来。

步骤4 执行。

在 DOS 状态下,直接键入文件名,可将可执行文件从磁盘调入内存并立即执行。在 Windows 状态下,直接双击可执行文件名即可使之运行。

步骤5 调试。

用调试程序,如 DEBUG、TDEBUG 等,调试可执行文件,检查是否有逻辑错误。通常一个较复杂的汇编语言源程序一次通过的可能性是很小的,免不了会出现一些意想不到的错误。遇到程序语法之外的错误,如算法等错误时,最好使用调试程序对可执行文件进行调试,以便方便地找到错误的根源。根据调试提供的信息,修改源程序文件并重新进行汇编链接过程。直到程序既无语法错误,又无逻辑错误为止。

以生成 MYFILE. EXE 文件为例,汇编、链接、调试的过程如图 2.1 所示。

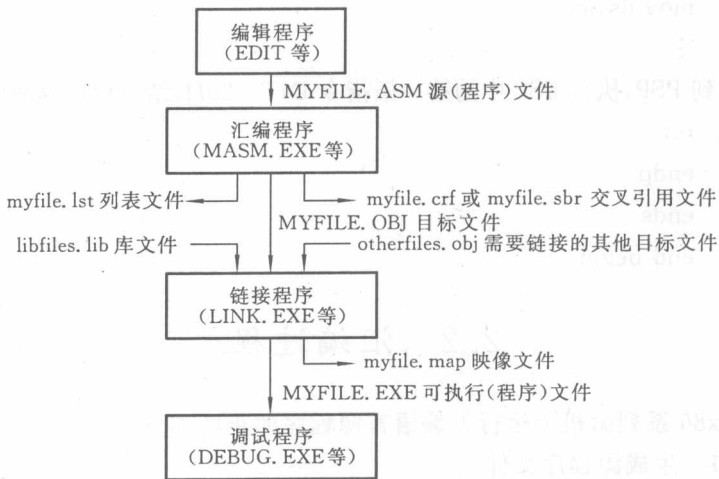


图 2.1 汇编语言源程序汇编、链接、调试的过程

2.3 用 EDIT 建立源程序文件

EDIT 是 DOS 操作系统自带的一个文本文件编辑器。文本文件是相对于

二进制文件而言的,文件内容由 ASCII 码字符构成。汇编语言源程序文件本质上是文本文件,只不过具有特定的文件后缀“. ASM”而已。使用 EDIT 可以完成 DOS 环境下汇编语言源程序的编写。EDIT 的界面如图 1.1 所示。

EDIT 的主菜单可以使用 F10 键或 Alt 键+<菜单项首字母>键来选取。例如:按 Alt-F 进入屏幕上端的 File 栏,回车后显示下拉菜单,移动光标可选择相应功能。

常用的主菜单命令及含义如下。

1. File→New

生成一个新文本文件。缺省文件名为“UNTITLEDx”,具体的文件名可以在保存时指定。

2. File→Open

打开一个已经存在的文本文件。对话框界面如图 2.2 所示。

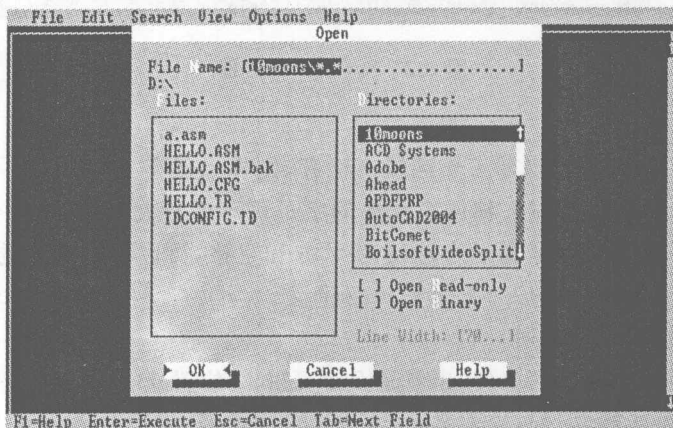


图 2.2 Open 菜单对话框

可以直接在 File Name 处输入要打开的文件名,也可在其下面的选择框内选择要打开的文件。按 Tab 键可以在不同的选择框之间切换。Directories 的下面列有驱动器名称,选择它们可以改变当前驱动器。

3. File→Save

保存文本文件。如果当前工作区内的文件已经命名,则直接保存;否则会弹出 Save As 对话框,如图 2.3 所示。其中可以指定保存文件名和文件路径。汇编语言源程序必须以. ASM 为后缀名。

4. File→Save As...

将当前工作区内的文件另存为其他文件,其对话框如图 2.3 所示。

5. File→Close

关闭当前工作文档。如果没有保存,则提示保存文件。

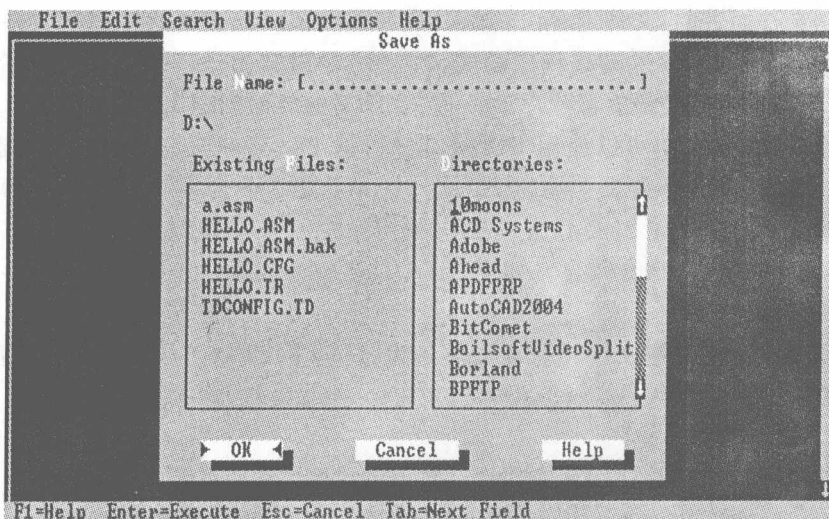


图 2.3 Save As 对话框

6. File→Exit

退出 EDIT。如果当前工作区没有保存,则提示保存文件。

需要指出的是,虽然汇编语言程序的字母可不区分大小写,对指令的缩进位置也不做明确要求,但是通过合理使用 Tab 和空格,可以形成良好的书写风格,有利于程序的读取与维护。下面是几条建议。

- 标号放置在第 1 列。
- 操作码放置在第 17 列,也就是 2 个 Tab 的位置。
- 操作数在第 25 列,也就是 3 个 Tab 的位置。
- 注释放置在第 41 或 49 列,也就是 5 个或 6 个 Tab 的位置。
- 如果注释需要占用 1 行,则从第 1 列开始。

可参照本书 1.1 中的 HELLO.ASM 源程序文件。

2.4 宏汇编程序 MASM

汇编语言源程序必须经过汇编,将源程序中的符号、符号指令和伪指令变为机器指令。该汇编工作可以用 Microsoft 宏汇编程序 MASM 来完成。

MASM 的语法为

```
masm /options source(.asm),[out(.obj)],[list(.lst)],[cref(.sbr)][;]
```

其中,“/options”为选项;“source”为编译对象,即需要编译的汇编源文件名;“out”为输出的目标文件名;“list”为列表文件名;“cref”为交叉引用文件名。目