

普通高等学校计算机科学与技术应用型规划教材

汇编语言程序设计案例式 实验指导

HUIBIAN

YUYAN CHENGXU SHEJI ANLISHI SHIYAN ZHIDAO

赵梅 杨永生 编著



北京邮电大学出版社
www.buptpress.com

普通高等学校计算机科学与技术应用型规划教材

汇编语言程序设计 案例式实验指导

赵 梅 杨永生 编 著



北京邮电大学出版社
[www. buptpress. com](http://www.buptpress.com)

内 容 简 介

全书共分5章。其中第1章简要介绍了汇编语言实验环境及上机过程。第2章包括8个基础性实验,每个实验要求学生自己动手独立完成实验内容,每个实验都给出了相近的案例,在案例中有较详细步骤指导,从而使学生做每一个实验都有参考。第3章包括4个综合性实验,每个实验内容都给出了分析与提示,以及可参考的运行结果。第4章介绍了4个演示程序,演示程序综合运用了汇编语言知识,展示了汇编语言的功能。第5章详细介绍了汇编语言的调试工具 debug。

全书实验内容丰富,伸缩性大,便于教学选择。附录中列出 DOS 系统功能调用、BIOS 调用和 ASCII 码表。

本书可作为大学本、专科计算机原理,汇编语言程序设计等相关课程实验实训的通用性教材。

图书在版编目(CIP)数据

汇编语言程序设计案例式实验指导/赵梅,杨永生编著.--北京:北京邮电大学出版社,2011.12

ISBN 978-7-5635-2857-8

I. ①汇… II. ①赵…②杨… III. ①汇编语言—程序设计 IV. ①TP313

中国版本图书馆 CIP 数据核字(2011)第 259272 号

书 名: 汇编语言程序设计案例式实验指导

著作责任者: 赵 梅 杨永生 编著

责任编辑: 王丹丹

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发行部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京联兴华印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 8

字 数: 159 千字

版 次: 2011 年 12 月第 1 版 2011 年 12 月第 1 次印刷

ISBN 978-7-5635-2857-8

定 价: 16.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

汇编语言程序设计是计算机科学与技术等专业的必修课程。近几年人们经常认为汇编语言的应用范围很小,而忽视它的重要性。其实汇编语言对每一个希望学习计算机科学与技术的人来说都是非常重要的,是不能不学习的语言。通过学习和使用汇编语言,能够感知、体会、理解机器的逻辑功能,向上可以理解各种软件系统,向下能够感知硬件,是我们理解整个计算机系统的最佳起点和最有效途径。因而汇编语言的实验环节就显得尤为重要。本教材旨在帮助学生加深认识和理解理论教学知识,通过大量的上机实验熟悉 8086 CPU 的指令功能、用途和使用技巧,进而提高汇编语言程序设计的能力。

本书以 Intel 8086 指令系统和 masm 为主体,在 PC 的 MS-DOS 和 Windows 操作系统平台中,使用 Edit 编辑软件,依据循序渐进的原则展开实验。其实目前汇编语言的编程环境很多,有很多使用的是集成环境,我们认为学习汇编语言就是要接触底层,要保持它的“原汁原味”,就是要 Edit、masm、link 一步一步地体会低级语言的上机全过程。

本书共分 5 章。其中第 1 章简要介绍了汇编语言实验环境及上机过程。第 2 章包括 8 个基础性实验,每个实验都给出了相近的案例,在案例中有较详细步骤指导。同时也给出了要求学生自己动手独自完成的实验内容。从而使学生做每一个实验都有参考。第 3 章包括 4 个综合性实验,每个实验内容都给出了分析与提示,以及可参考的运行结果。第 4 章介绍了 4 个演示程序,程序综合地运用了汇编语言知识展示了汇编语言的功能。第 5 章详细地介绍了汇编语言的调试工具 debug。

本书特点是力求简明实用,既有步骤指导又要求学生自己动手独立完成。所以书中的基础实验部分采用了案例式实验方法指导学生实验,也就是要求学生独立完成的实验,都给出了相近的案例;在案例中给出了详细的操作步骤,从而使学生能够自己动手独立做实验;实验中有参考,以适应培养应用型本科生的教学需求。同时也体现了认知上的循序渐进,突出实验原理、实验方法,便于学生对实验原理的理解。在综合实验中强调了汇编语言的模块化程序设计的思想。

本书由赵梅、杨永生编写,书中的全部程序都经过了调试和运行。书中如有错误和不当之处,欢迎读者批评指正。

编 者

目 录

第 1 章 汇编语言实验环境及上机过程	1
1.1 全屏幕编辑程序——Edit	1
1.1.1 Edit 的功能及应用范围	1
1.1.2 Edit 的启动与退出	1
1.1.3 编辑一个汇编语言源程序	2
1.2 汇编语言调试程序——debug 简介	3
1.2.1 debug 的主要特点	3
1.2.2 通过 debug 编写\运行汇编语言程序	4
1.3 宏汇编程序(masm)及汇编语言上机过程	6
1.3.1 运行汇编语言程序必备的条件	6
1.3.2 执行宏汇编程序	7
1.3.3 执行连接程序	8
1.3.4 执行程序	10
第 2 章 基础实验	11
2.1 实验一 寻址方式练习	11
2.1.1 寻址方式实验案例	11
2.1.2 自己动手做寻址方式练习实验	15
2.2 实验二 多字节加法实验	15
2.2.1 多字节加法实验案例	15
2.2.2 自己动手做多字节加法实验	19
2.3 实验三 顺序程序设计	20
2.3.1 字符串查找实验案例	20
2.3.2 自己动手做字符串查找与替换实验	24
2.4 实验四 分支程序设计 1	24
2.4.1 统计成绩实验案例	24

2.4.2 自己动手做统计字符实验	29
2.5 实验五 分支程序设计 2	30
2.5.1 键盘输入数据保存到寄存器中实验案例	30
2.5.2 自己动手做键盘输入数据保存到存储单元实验	34
2.6 实验六 循环程序设计 1	34
2.6.1 字符组成矩形输出实验案例	34
2.6.2 自己动手做平行四边形输出实验	39
2.7 实验七 循环程序设计 2	40
2.7.1 ASCII 码转换实验案例(小写字母转换大写字母输出)	40
2.7.2 自己动手完成 ASCII 码转换实验(大写转换为小写)	44
2.8 实验八 子程序设计	44
2.8.1 数制转换实验案例(键盘输入十六进制数,输出二进制数)	44
2.8.2 自己动手做数制转换实验(输入十进制数,输出二进制数)	50
第 3 章 综合性实验	51
3.1 综合实验报告	51
3.2 实验一 明文密文转换实验	52
3.1.1 实验目的与要求	52
3.1.2 实验内容	52
3.1.3 分析与提示	52
3.3 实验二 子程序嵌套实验	53
3.3.1 实验目的与要求	53
3.3.2 实验内容	53
3.3.3 分析与提示	54
3.4 实验三 计算有符号数的和、差、积、商及余数	55
3.4.1 实验目的与要求	55
3.4.2 实验内容	56
3.4.3 分析与提示	56
3.5 实验四 输出通讯录实验	60
3.5.1 实验目的与要求	60
3.5.2 实验内容	60
3.5.3 分析与提示	60

第 4 章 汇编语言演示程序	63
4.1 “枪声”程序	63
4.1.1 源程序	63
4.1.2 程序运行结果	64
4.2 “小车移动”程序	65
4.2.1 源程序	65
4.2.2 程序运行结果	69
4.3 打字计时程序	69
4.3.1 源程序	70
4.3.2 程序运行结果	79
4.4 演奏音阶程序	79
4.4.1 源程序	80
4.4.2 程序运行结果	81
第 5 章 汇编语言程序调试工具 debug	82
5.1 debug 调试工具	82
5.1.1 debug 的启动	82
5.1.2 debug 的主要命令	83
5.2 用 debug 调试 .exe 文件	99
附录 I DOS 系统功能调用	103
附录 II BIOS 功能调用	111
附录 III ASCII 码字符表	115
附录 IV 汇编程序出错信息	116
参考文献	120

第 1 章 汇编语言实验环境及上机过程

本章介绍的汇编语言实验是基于 DOS 平台,如果使用 Windows 平台,可转换到 MS-DOS 程序方式,即建立 DOS 平台。

1.1 全屏幕编辑程序——Edit

在编写汇编语言源程序时,可通过全屏幕编辑程序 Edit 或 PE 等编辑软件建立源程序。

1.1.1 Edit 的功能及应用范围

Edit 是基于 DOS 平台的文本编辑工具,具有常规的操作命令和灵活的键盘功能。其主要功能如下:

- (1) 具有丰富的编辑命令。
- (2) 可以调入保存文件进行编辑,可以对字符串进行查找、替换等操作。
- (3) 具有区域标记定义和操作功能。
- (4) 可以进行区块的定义、剪切、复制、删除等操作,可以设置前景及背景颜色。
- (5) 灵活的键盘功能。
- (6) 方便的打印功能。

Edit 编辑软件,如同我们常用的 Word 软件一样有文件(File)菜单、编辑(Edit)菜单等。使用方法也很类似。

文件菜单中有新建(New)、打开(Open…)、保存(Save)、另存为(Save As…)、打印(Print…)、退出(Exit)等功能。

编辑菜单中有剪切(Cut)、复制(Copy)、粘贴(Paste)、清除(Clear)等功能。

下面就以编写、保存一个汇编语言源文件为例,简单介绍 Edit 软件的使用。

1.1.2 Edit 的启动与退出

1. 启动

Edit 的启动同其他 DOS 外部命令一样,只需在 DOS 状态下输入

edit [文件名]

其中[文件名]是一个任选项,如果指定文件名,Edit 就装入指定的文件以供编辑。如果不指定文件,则 Edit 就建立一个无名文件,其后用户可在存盘时指定文件名。

2. 存盘与退出

按 Alt 键,激活菜单(或直接单击菜单栏),选择 File 菜单,再选择 Save...或 Save As ...或 Exit。分别为保存、另存为和退出。

1.1.3 编辑一个汇编语言源程序

在 DOS 状态下可以编辑一个汇编语言源程序,具体步骤如下:

(1) 在 Windows 操作系统下选择“开始”→“程序”→“附件”→“命令提示符”,进入 DOS 环境。

(2) 在 DOS 状态下进入 Edit 状态。如进入 d:盘 masm 目录,输入命令,如图 1-1 所示,输入:edit lx.asm。

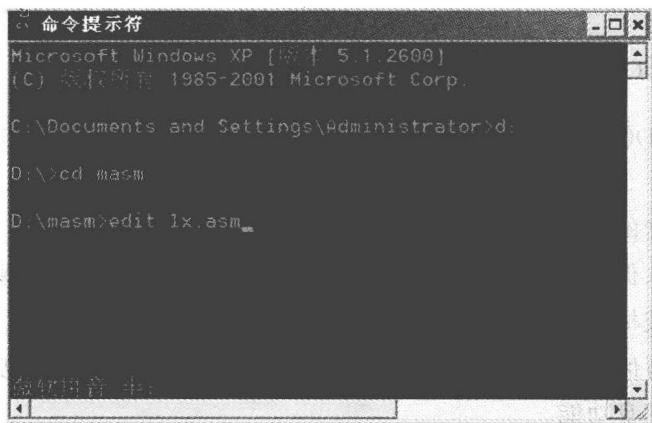


图 1-1 DOS 窗口

(3) 在编辑(Edit)状态下输入用汇编语言编写的源程序 lx.asm,如图 1-2 所示。

(4) 保存源文件。按 Alt 键激活菜单,选择文件(File)菜单,选择 Save 命令保存源文件。即将图 1-2 所示的汇编语言源文件以文件名 lx.asm 保存到当前盘当前目录。

(5) 退出 Edit 状态。选择 File 菜单,选择 Exit 命令,退出 Edit 状态,返回到 DOS 状态。

至此,我们使用 Edit 编辑程序编写了一个汇编语言源文件 lx.asm,并保存到了当前路径下。



图 1-2 编辑程序

1.2 汇编语言调试程序——debug 简介

在编写和运行汇编语言程序的过程中,会出现一些错误和问题,需要对程序进行分析和调试,调试程序 debug 就是专为小汇编和宏汇编语言设计的一种调试工具。它在调试汇编语言程序时有很强的功能,能使程序设计者接触到机器内部,观察和修改寄存器和存储单元内容,并监视目标程序的执行情况,使用户真正接触到 CPU 内部,与计算机产生最紧密的工作联系。关于 debug 命令的具体使用将在第 5 章详细介绍。这里我们先介绍一下它的主要特点。

1.2.1 debug 的主要特点

1. 能够在最小环境下运行汇编程序

在 DOS 状态下运行汇编程序,必须将程序经过 asm 或 masm 汇编后,再经过 link 连接程序产生可执行程序,才能最终运行。而在 debug 状态下,为用户提供了调试、控制测试的环境,可以在此环境下进行编程、调试、监督、执行用户编写的汇编程序。因此调试周期短,为用户提供了极大的方便。

2. 提供极简单的修改手段

debug 提供了修改命令,可以修改内存单元内容,为调试程序、修改程序带来了方便。

3. 提供用户与计算机内部联系的窗口

debug 具有显示命令,它既可以使用户看到某内存单元或某一块单元内容,也可以看到 CPU 内部各寄存器的内容。可用单步执行命令跟踪执行,每执行一步都能使用户看

到各寄存器的内容变化,以便分析和调整程序。

4. 可装入、修改或显示文件

当然在 debug 状态下运行汇编语言程序也具有一定局限性:如在 debug 状态下运行的程序不能使用宏汇编程序中的宏指令,大部分伪指令也不能使用,因此只能把程序分段调试。

其次,不能调试太长的程序,只能分块进行程序设计,而且在 debug 状态下调试好的程序只能形成可执行文件(.exe),因此调试好的只能记下,到编辑环境下输入调试好的程序,通过汇编程序(asm 或 masm),再通过连接程序(link)形成可执行文件(.exe)。

1.2.2 通过 debug 编写/运行汇编语言程序

下面通过例子使大家了解在 debug 状态下编写/运行汇编语言程序的过程。

例 1.1 用汇编语言编写一个简单的加法,程序段如下:

```
mov al,33h ;将 3 的 ASCII 码送 al 寄存器中
mov dl,35h ;将 5 的 ASCII 码送 dl 寄存器中
add dl, al ;3+5 结果送 dl 寄存器中
sub dl, 30h;调整结果,得到 8 的 ASCII 码,送 dl 寄存器
mov ah,2 ;
int 21h ;输出 dl 寄存器中的字符
int 20h ;中断当前执行程序
```

该程序前两条指令是将两个十进制数 3 和 5 的 ASCII 码送入寄存器 al 和 dl。第 3 条语句是做加法,当 ASCII 码进行相加时必须通过调整后才能得到其结果。如例 1.1 中,33h+35h=68h,68h-30h=38h,38h 才是十进制 8 的 ASCII 码。所以第 4 条语句减法,减 30h,然后将结果在显示器上输出,这里 int 21h 为 DOS 系统功能调用,功能号为 2,该功能是在显示器上显示 dl 寄存器中的字符(即字符的 ASCII 码放在 dl 寄存器中)。int 20h 是正常结束程序。

程序输入及运行步骤如下:

(1) 进入 debug 状态。

在 DOS 状态下,进入装有 debug 程序的磁盘目录,如 d 盘 masm 目录下装有 debug 调试工具。

```
d:\>cd masm ✓
```

```
d:\masm\>debug ✓
```

屏幕显示:

—

“—”为 debug 的提示符,表明进入了 debug 状态,在该提示符下可输入 debug 命令。以下下画线部分为用户输入的字符或命令。

(2) 输入程序并汇编。

用 debug 的 a 命令送入程序:

```
—a100 ✓
    0A47:0100 mov al,33 ✓
    0a47:0102 mov dl,35 ✓
    0a47:0104 add dl,al ✓
    0a47:0106 sub dl,30 ✓
    0a47:0109 mov ah,2 ✓
    0a47:010b int 21 ✓
    0a47:010d int 20 ✓
    0a47:010f ✓
—
```

当输入 a 命令(汇编命令)时,会自动产生程序所送内存单元的段地址和偏移地址,通过偏移地址可以看到每条指令占内存单元多少个字节,mov al,33 占两个字节,sub dl,30 占 3 个字节。当程序段送完时,只输入回车键,就退出汇编状态(a 命令状态)回到 debug 状态“—”。其中送入数据为十六进制数,debug 状态下程序中的数据均按十六进制处理,不需要输入 h 来表示数据为十六进制。

(3) 执行程序。

用 debug 的 g 命令执行刚刚汇编的程序:

```
-g ✓
8
Program terminated normally
```

显示的 8 就是程序运行的结果($3+5=8$)。

(4) 反汇编。

用反汇编 u 命令将输入的程序反汇编(显示)出来,并且可以得到每条汇编指令的机器码。

```
-u 100 10d ✓
0a47:0100 b033      mov  al,33
0a47:0102 b235      mov  dl,35
0a47:0104 00c2      add  dl,al
0a47:0106 80ea30     sub  dl,30
```

```
0a47:0109 b402      mov  ah,02
0a47:010b cd21      int  21
0a47:010d cd20      int  20
```

其中 u 命令后面的地址为要反汇编程序的起始偏地址和终止偏地址。

(5) 退出 debug, 返回 DOS 状态。

```
-q ✓
d:\masm>
```

至此, 完成了在 debug 下输入程序和运行程序。debug 的命令很多很丰富, 我们在第 5 章展开并举例介绍。

1.3 宏汇编程序(masm)及汇编语言上机过程

汇编就是把用汇编语言编写的源程序翻译(汇编)成机器语言的目标程序。

汇编程序可以是小汇编程序(asm)也可以是宏汇编程序(masm), 由于宏汇编程序不但可以替换 asm, 而且可以汇编具有宏定义的汇编语言源程序, 因此我们在汇编程序时使用宏汇编程序(masm)。

1.3.1 运行汇编语言程序必备的条件

在当前盘上一般要包括如下文件:

- masm——宏汇编程序。
- link——连接程序。
- cref——索引程序(也可不用)。
- edit——全屏幕文本编辑程序(或其他文本编辑文件)。
- debug——调试工具。

用户通过屏幕编辑程序(如 Edit)输入源程序, 检查无误, 该源程序的扩展名必须为 .asm。能用宏汇编检查通过的汇编语言源程序与在 debug 状态下运行的汇编语言程序不同, 必须是一个完整的程序, 有各逻辑段的定义, 而在 debug 状态下直接汇编运行的汇编语言程序只是其程序段。

下面是在编辑状态下编辑的汇编语言源程序。该程序的功能是在屏幕上显示数据段中的一串英文字符 'how are you!'. 程序如下, 我们为它命名为 ex.asm。

```
data segment
    s1 db 'how are you!', '$'
data ends
```

```

stack segment stack
    db 64 dup(?)
stack ends
code segment
    assume cs:code,ds:data
start: mov ax,data
    mov ds,ax
    mov ah,09h
    mov dx,offset s1
    int 21h
    mov ah,4ch
    int 21h
code ends
end start

```

1.3.2 执行宏汇编程序

将汇编语言源程序用宏汇编程序翻译(汇编)后,可以形成三个文件:一个是扩展名为.obj的目标文件,在该文件中将源程序的操作码部分变为机器码,但地址操作数是可浮动的相对地址,而不是实际地址,因此需要经link连接文件进行连接才能形成可执行文件。第二个文件是列表文件,扩展名为.lst,它把源程序和目标程序列表,以供检查程序用。第三个文件是交叉索引文件,扩展名为.crf,它是一个对源程序所用的各种符号进行前后对照的文件。其中目标文件是必须产生的,而其他两个文件在需要时给予命令就可产生,对连接和执行汇编语言程序无直接的关系。汇编过程如下。

在DOS状态下,输入masm_↵则调入宏汇编程序,屏幕显示与操作如下:

```

d:\masm>masm↵
Microsoft (R) Macro Assemble Version 5.00
Copyright (C) Microsoft Corp 1981-1985,1987,All rights reserved
Source filename [.ASM]:ex↵
Object filename [EX.OBJ]:ex↵
Source listing[NUL.LST]:ex↵
Cross-reference[NUL.CRF]:ex↵

51612 + 422724 Bytes symbol space free

0 Warning Errors

```

0 Severe Errors

其中划线部分为用户输入部分,ex 为源程序名(ex.asm),方括号中是机器规定的默认文件名,如果用户认为方括号内的文件名就是要输入的文件名,则可只在划线部分直接输入回车键。如果不需要生成列表文件和交叉文件,则可在[nul.lst]和[nul.crf]后不输入文件名,只输入回车键。

当回答完上述四个询问后,汇编程序就对源程序进行汇编。在汇编过程中,如果发现源程序中有语法错误,则提示出错信息,指出是什么性质的错误,错误类型,最后列出错误的总数。之后,用户可重新进入编辑程序,调入源程序(ex.asm)进行修改,修改完毕,再进行汇编。汇编时用分号结束,只产生目标文件。例如:

```
d:\masm>masm ex;↵
```

Microsoft (R) Macro Assemble Version 5.00

Copyright (C) Microsoft Corp 1981-1985,1987,All rights reserved

51612 + 422724 Bytes symbol space free

0 Warning Errors

0 Severe Errors

汇编后只产生一个.obj 文件。如果需要产生.obj 文件和.lst 文件,不需要.crf 文件,则在分号前加两个逗号即可。

例如:

```
d:\masm>masmex , , ;↵
```

如果 4 个文件都需要,用简便的方法来操作,操作方法如下:(分号前面用了 3 个逗号)

```
d:\masm>masm ex,,,;↵
```

Microsoft (R) Macro Assemble Version 5.00

Copyright (C) Microsoft Corp 1981-1985,1987,All rights reserved

51612 + 422724 Bytes symbol space free

0 Warning Errors

0 Severe Errors

1.3.3 执行连接程序

用汇编语言编写的源程序经过汇编程序(masm)汇编后产生了目标程序(.obj),该文件是将源程序操作码部分变成了机器码,但地址是浮动的相对地址(逻辑地址),因此必须经过连接程序 link 连接后才能运行。连接程序 link 是把一个或多个独立的目标程序模块装配成一个可重定位的可执行文件,扩展名为.exe 文件。此外还可以产生一个内存

镜像文件,扩展名为.map。

1. 连接程序执行过程

在DOS状态下,输入link (或link ex)则系统调入link程序,屏幕显示操作如下:

```
d:\masm>link
Microsoft Macro Assembler Version 5.00
Copyright Microsoft Corp 1981-1985,1987.All rights reserved
Object Modules[.OBJ]:ex
Run File [EX.EXE]:ex
List File [UNL.MAP]:ex
Libraries [.LIB]:
```

其中划线部分为用户输入部分,ex为源程序名,方括号内为机器默认文件名,当用户认为方括号中的文件名就是要输入的文件名时,可在冒号后面只键入回车键。

其中.map文件是否需要建立,由用户决定,需要则输入文件名,不需要则直接输入一个回车键。

最后一个询问是是否在连接时用到库文件,对于连接汇编语言源程序的目标文件,通常是不需要的,因此直接输入回车键。

与汇编程序一样,可以在连接时用分号结束后续询问。

例如:

```
d:\masm\>link ex;
Microsoft Macro Assembler Version 5.00
Copyright Microsoft Corp 1981-1985,1987.All rights reserved
```

连接后只产生ex.exe文件。如果除ex.exe文件外还要产生ex.map文件,则在分号前加两个逗号。

例如:

```
d:\masm\>link ex,,;
Microsoft Macro Assembler Version 5.00
Copyright Microsoft Corp 1981-1985,1987.All rights reserved
```

2. 内存镜像文件(.amp)

由连接程序link产生的扩展名为.map文件,它实际上是连接程序的列表文件,它给出了每个段的地址分配情况及长度。

在DOS状态下,用type命令显示打印出来。具体方法如下:

```
d:\masm>type ex.map
```

start	stop	length	name	class
00000	0000ch	0000dh	data	
00010h	0001fh	00010h	code	
00020h	0005fh	00040h	stack	

Program entry point at 0001:0000

从表中可以看到,源程序 ex 中定义了三个段:数据段(data)起始地址为 00000h,终止地址为 0000ch,长度为 0dh 个字节;代码段(code)起始地址为 00010h,终止地址为 0001fh,长度为 10h 个字节;堆栈段(stack)起始地址为 00020h,终止地址为 0005fh,长度为 40h 个字节。

➤ 1.3.4 执行程序

当用连接程序 link 将目标程序(.obj)连接定位后,产生可执行文件(.exe),在 DOS 状态下执行该程序。

执行操作如下:

d:\masm>ex ↵

how are you !

也可输入 ex.exe ↵

d:\masm>ex.exe ↵

how are you !

在执行程序后可以看到执行的结果,是因为源程序中有显示结果的指令。有的程序没有显示结果指令,要想看到结果,只有通过 debug 调试程序达到目的。如果执行结果没有达到预先设计目的,也是通过 debug 来进行调试、运行。因此 debug 是汇编语言程序的最有利的调试工具。