

全国高职高专规划教材

汇编语言程序设计 实训教程

Programming Assembly
Language in Practice

陈伟芳 主 编
倪步喜 黄建设 副主编



科学出版社

www.sciencep.com



全国高职高专规划教材

汇编语言程序设计实训教程

陈伟芳 主 编

倪步喜 黄建设 副主编

科 学 出 版 社

北 京

内 容 简 介

本书主要阐述了 Intel 8086/8088 系列汇编语言程序设计及汇编语言与 C 语言混合编程的基本方法。全书共分 11 章:第 1 章介绍汇编语言调试工具 DEBUG;第 2 章介绍 Intel 8086/8088 的指令系统使用及调试方法;第 3 章介绍汇编语言源程序的书写格式与顺序程序设计及其调试方法;第 4、5、6 章分别介绍分支、循环、子程序的程序设计及调试方法;第 7 章介绍串指令的应用和调试;第 8 章说明中断服务程序的编写方法;第 9、10 章介绍 DOS 和 BIOS 中断的调用方法;第 11 章介绍汇编语言与 C 语言交叉汇编的方法。

本书以实践为主,通过实践加强理论的理解,适合初学者,也可作为高职高专教材。

图书在版编目(CIP)数据

汇编语言程序设计实训教程/陈伟芳主编. —北京:科学出版社, 2004

(全国高职高专规划教材)

ISBN 7-03-012629-7

I. 汇... II. 陈... III. 汇编语言—程序设计—高等学校:技术学校—教材
IV. TP313

中国版本图书馆 CIP 数据核字(2003)第 120922 号

策划编辑:李振格/责任编辑:熊盛新

责任印制:吕春珉/封面制作:东方人华平面设计室

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004 年 1 月第 一 版 开本: 787×1092 1/16

2004 年 1 月第一次印刷 印张: 13 3/4

印数: 1—5 000 字数: 308 000

定价: 19.00 元

(如有印装质量问题,我社负责调换(路通))

全国高职高专规划教材编委会名单

主 任 俞瑞钊

副主任 陈庆章 周必水 刘加海

委 员 (以姓氏笔画为序)

王 雷 王筱慧 方 程 方锦明 卢菊洪 代绍庆

吕何新 朱 炜 刘向荣 江爱民 江锦祥 孙光弟

李天真 李永平 李良财 李明钧 李益明 余根墀

汪志达 沈凤池 沈安衢 张 元 张学辉 张锦祥

张德发 陈月波 陈晓燕 邵应珍 范剑波 欧阳江林

周国民 周建阳 赵小明 胡海影 秦学礼 徐文杰

凌 彦 曹哲新 戚海燕 龚祥国 章剑林 蒋黎红

董方武 鲁俊生 谢 川 谢晓飞 楼 丰 楼程伟

鞠洪尧

秘书长 熊盛新

本书编写人员名单

主 编 陈伟芳

副主编 倪步喜 黄建设

撰稿人 陈伟芳 倪步喜 黄建设 秦本涛

前 言

本教材以 IBM-PC 系列机为背景,阐述了 Intel 8088/8086 系列的汇编语言指令和汇编语言程序设计。本教材强调基础,突出应用,以大量的例子来讲述汇编语言指令、编程方法及调试方法,充分体现计算机最底层的程序设计方法。

汇编语言是运行速度最快,并能直接接触计算机硬件的一种程序设计语言,也是计算机专业学生的核心课程之一。本书在编写过程中,充分考虑到汇编语言的特点,尽量浅化理论知识,强化实践操作能力,始终以实践操作为主线,并通过实践来巩固理论知识的学习,达到理论与实践并进的目的。

本书编写的几点说明:

1. 本书中的例子均在 DOS 平台或 Windows 9X 下的 MS-DOS 方式下完成,建议读者也在该平台下操作。

2. 为了与 DEBUG 中的表示形式一致,本书中表示寄存器内容一律不加括号,如叙述中 AX=1234H,表示 AX 寄存器内容为 1234H。

3. 由于汇编语言本身对字母的大小写不敏感,为了便于教与学,本书中一律采用大写的形式。

4. 带下划线表示从键盘输入的命令,用↵表示输入回车键。

本书由浙江树人大学陈伟芳任主编,温州职业技术学院倪步喜、浙江工商职业技术学院黄建设任副主编,衢州浙工大浙西分校秦本涛任参编。陈伟芳编写第 2 章(部分)、第 3 章、第 4 章、第 6 章、第 10 章、第 11 章和附录;倪步喜编写第 7 章、第 8 章、第 9 章;黄建设编写第 1 章、第 2 章(部分);秦本涛编写第 5 章。全书由陈伟芳统稿。

本书适合计算机初学者使用,具有高级语言程序设计基础的读者,都可以通过本书掌握汇编语言程序设计的方法。本书是以实训为主的教材,也可作为理论学习的实验指导用书。

由于时间仓促,书中难免有错误和不妥之处,真诚希望广大读者批评指正。

编 者

2003 年 7 月

目 录

第1章 调试程序 DEBUG	1
1.1 实训概要	1
1.1.1 调试程序 DEBUG 简介	1
1.1.2 DEBUG 命令	3
1.2 实训案例操作分析	14
1.3 实训项目一 DEBUG 常用命令的使用	16
1.3.1 实训目的	16
1.3.2 实训操作及流程分析	16
1.3.3 实训操作步骤	18
1.4 实训项目二 传送指令程序段的调试	20
1.4.1 实训目的	20
1.4.2 实训操作及流程分析	21
1.4.3 实训操作步骤	22
小结	24
习题	24
第2章 Intel 8086/8088 的指令系统	25
2.1 实训概要	25
2.1.1 数据传送指令	25
2.1.2 算术运算类指令	28
2.1.3 逻辑运算类指令	35
2.1.4 处理器控制指令	37
2.2 实训案例操作分析	38
2.3 实训项目一 数据传送指令的实训	40
2.3.1 实训目的	40
2.3.2 实训操作及流程分析	40
2.3.3 实训操作步骤	41
2.4 实训项目二 加、减指令的运用	41
2.4.1 实训目的	41
2.4.2 实训操作及流程分析	42
2.4.3 实训操作步骤	42
2.5 实训项目三 乘、除指令的运用	43
2.5.1 实训目的	43
2.5.2 实训操作及流程分析	43

2.5.3 实训操作步骤	44
2.6 实训项目四 逻辑运算指令的运用	44
2.6.1 实训目的	44
2.6.2 实训操作及流程分析	45
2.6.3 实训操作步骤	45
2.7 实训项目五 BCD 码的操作	46
2.7.1 实训目的	46
2.7.2 实训操作及流程分析	46
2.7.3 实训操作步骤	47
小结	47
习题	48
第3章 顺序程序设计	49
3.1 实训概要	49
3.1.1 语句	49
3.1.2 汇编语言中的数据	50
3.1.3 程序的段结构	54
3.2 实训案例操作分析	57
3.3 实训项目一 顺序程序设计	58
3.3.1 实训目的	58
3.3.2 实训操作及流程分析	58
3.3.3 实训操作步骤	60
3.4 实训项目二 可执行文件的调试	61
3.4.1 实训目的	61
3.4.2 实训操作及流程分析	61
3.4.3 实训操作步骤	65
小结	66
习题	66
第4章 分支程序设计	68
4.1 实训概要	68
4.1.1 无条件转移指令	68
4.1.2 条件转移指令	69
4.1.3 转移指令构成的重复结构	71
4.2 实训案例操作分析	73
4.3 实训项目一 分支程序设计	74
4.3.1 实训目的	74
4.3.2 实训操作及流程分析	74
4.3.3 实训操作步骤	75
4.4 实训项目二 分支程序的调试	76

4.4.1 实训目的	76
4.4.2 实训操作及流程分析	76
4.4.3 实训操作步骤	80
小结	81
习题	81
第5章 循环程序设计	83
5.1 实训概要	83
5.2 实训案例操作分析	85
5.3 实训项目一 循环程序的设计	88
5.3.1 实训目的	88
5.3.2 实训操作及流程分析	88
5.3.3 实训操作步骤	94
5.4 实训项目二 循环程序的调试	96
5.4.1 实训目的	96
5.4.2 实训操作及流程分析	96
5.4.3 实训操作步骤	100
小结	102
习题	102
第6章 子程序的设计	104
6.1 实训概要	104
6.1.1 过程的定义	104
6.1.2 调用与返回指令	106
6.1.3 编制过程的基本要求	107
6.2 实训案例操作分析	108
6.3 实训项目一 子程序的设计	110
6.3.1 实训目的	110
6.3.2 实训操作及流程分析	110
6.3.3 实训操作步骤	114
6.4 实训项目二 子程序的调试	116
6.4.1 实训目的	116
6.4.2 实训操作及流程分析	116
6.4.3 实训操作步骤	119
小结	121
习题	122
第7章 串处理指令	123
7.1 实训概要	123
7.1.1 串操作指令	123
7.1.2 重复前缀指令	124

7.2 实训案例操作分析	126
7.3 实训项目一 串处理程序的设计	129
7.3.1 实训目的	129
7.3.2 实训操作及流程分析	129
7.3.3 实训操作步骤	129
7.4 实训项目二 串处理程序的调试	130
7.4.1 实训目的	130
7.4.2 实训操作及流程分析	130
7.4.3 实训操作步骤	130
小结	130
习题	130
第 8 章 中断服务程序的编制	132
8.1 实训概要	132
8.1.1 一般概念	132
8.1.2 中断源及分类	132
8.1.3 中断类型码和中断向量表	134
8.1.4 中断优先级	135
8.1.5 中断处理过程	135
8.1.6 编制用户自定义中断处理程序	136
8.2 实训案例操作分析	143
8.3 实训项目一 中断向量表的结构	148
8.3.1 实训目的	148
8.3.2 实训操作及流程分析	148
8.3.3 实训操作步骤	148
8.4 实训项目二 中断服务程序的编制	150
8.4.1 实训目的	150
8.4.2 实训操作及流程分析	150
8.4.3 实训操作步骤	150
小结	150
习题	150
第 9 章 DOS 功能调用	151
9.1 实训概要	151
9.1.1 字符串操作	151
9.1.2 DOS 文件操作	153
9.2 实训案例操作分析	162
9.3 实训项目一 INT 21H 的基本操作	166
9.3.1 实训目的	166
9.3.2 实训操作及流程分析	167

9.3.3 实训操作步骤	167
9.4 实训项目二 字符串操作	168
9.4.1 实训目的	168
9.4.2 实训操作及流程分析	168
9.4.3 实训操作步骤	168
9.5 实训项目三 文件操作	169
9.5.1 实训目的	169
9.5.2 实训操作及流程分析	170
9.5.3 实训操作步骤	170
小结	171
习题	171
第 10 章 BIOS 功能调用	172
10.1 实训概要	172
10.2 实训案例操作分析	173
10.3 实训项目一 CMOS 的读写	175
10.3.1 实训目的	175
10.3.2 实训操作及流程分析	176
10.3.3 实训操作步骤	178
10.4 实训项目二 软盘的访问	179
10.4.1 实训目的	179
10.4.2 实训操作及流程分析	179
10.4.3 实训操作步骤	183
10.5 实训项目三 硬盘的访问	184
10.5.1 实训目的	184
10.5.2 实训操作及流程分析	184
10.5.3 实训操作步骤	186
10.6 实训项目四 主引导扇区的访问	186
10.6.1 实训目的	186
10.6.2 实训操作及流程分析	186
10.6.3 实训操作步骤	189
小结	189
习题	190
第 11 章 C 语言调用汇编程序	191
11.1 实训概要	191
11.2 实训案例操作分析	192
11.3 实训项目一 Turbo C 的基本操作	195
11.3.1 实训目的	195
11.3.2 实训操作及流程分析	195

11.3.3 实训操作步骤	196
11.4 实训项目二 C 程序调用汇编程序	197
11.4.1 实训目的	197
11.4.2 实训操作及流程分析	197
11.4.3 实训操作步骤	197
11.5 实训项目三 汇编和 C 程序的混合应用	198
11.5.1 实训目的	198
11.5.2 实训操作及流程分析	198
11.5.3 实训操作步骤	200
小结	201
习题	202
附录	203
附录 1 CMOS 数据格式	203
附录 2 ROM 信息	204
附录 3 分区类信息表	205
附录 4 EXE 文件头信息	205
附录 5 软、硬盘读写功能 (INT 13H)	205
附录 6 程序段前缀 (PSP) 结构	207
主要参考文献	208

第 1 章 调试程序 DEBUG

1.1 实训概要

通过本章的学习,熟悉调试程序 DEBUG 的应用环境,掌握 DEBUG 命令的基本格式及程序调试的基本方法,最后能达到熟练使用 DEBUG 的目的。本章实训的重点是 DEBUG 环境下的操作方法,并通过 DEBUG 命令对程序的调试,掌握 DEBUG 中几个较重要的命令,初步了解程序调试的基本方法。在进行本章的实训学习前,学生应对 MOV、ADD 等指令有初步的了解。

1.1.1 调试程序 DEBUG 简介

DEBUG.EXE 是 DOS 操作系统提供的一个调试汇编语言程序的工具程序,每个版本的 DOS 都带有该程序。利用 DEBUG 程序,程序员可动态调试汇编语言程序。程序员可以检查或修改 CPU 各寄存器、标志及内存单元的内容,也可以输入汇编程序指令程序到指定的内存单元,或把要调试的程序(通常以可执行文件存储在磁盘上)调入到内存中,控制 CPU 按单步或设置断点等方式执行程序,检查程序运行过程中的中间结果,以便寻找程序出错原因。

DEBUG 程序采用的是命令行方式,即通过键盘输入命令,按回车键后开始执行该命令。尽管这种方式不太方便,但由于 DEBUG 程序的实用性,仍使其成为学习汇编语言程序设计、计算机硬件技术等课程的有效工具。本节主要通过几个小案例来介绍 DEBUG 程序的使用。

1. 直接启动 DEBUG 程序

格式: DEBUG

功能: 该命令是在 DOS 下把 DEBUG.EXE 程序调入内存,并运行 DEBUG 程序。命令输入后,出现提示符“-”,表示已在 DEBUG 状态,可执行 DEBUG 程序的命令。DEBUG 相当于 DOS 的一个外部命令,其版本应与 DOS 版本相符,若两者的版本不同,输入 DEBUG 后屏幕上将显示出错信息“Incorrect DOS Version”,表示 DEBUG 的启动失败。

说明: DEBUG 调试程序是在 DOS 下运行的一个软件。如果使用的是 DOS 操作系统,在 DOS 的提示符“>”下输入 DEBUG;如果使用的是 Windows 平台,则可以使用 Windows 中的“MS-DOS 方式”来启动 DEBUG。

2. 启动 DEBUG 程序的同时装入被调试文件

命令格式如下:

DEBUG [盘符][路径]文件名.扩展名

如: ABC.EXE 可执行文件存放在 A 盘根目录, 用 DEBUG 调试时, 装入程序的操作命令如下:

```
>DEBUG A:\ABC.EXE✓
```

DOS 在调用 DEBUG 程序后, 再由 DEBUG 把被调试文件装入内存, 当被调试文件的扩展名为.COM 时, 装入偏移地址为 100H 的位置, 当扩展名为.EXE 时, 装入偏移地址为 0000H (因为.EXE 文件需要重定位, 默认的偏移地址为 0000H, 当用 ORG 指定时, 就采用指定值) 的位置, 并建立程序段前缀 PSP。

3. 启动 DEBUG 程序的同时装入带有参数的被调试文件

命令格式如下:

```
DEBUG [盘符][路径]文件名.扩展名 参数 1[参数 2]
```

如:

```
C:\>DEBUG DISKCOMP.COM A: B:✓
```

其中的 A: 与 B: 参数是 DISKCOMP.COM 文件所要求的。

4. 退出 DEBUG 程序

在 DEBUG 命令提示符 “-” 下键入 Q 命令, 即可结束 DEBUG 的运行, 返回 DOS 操作系统。

5. 在 DEBUG 环境下建立和汇编程序

在 DEBUG 环境下用户可以直接建立汇编语言源程序, 也可以进行编辑修改, 还可以进行汇编。

比如, 在 DEBUG 下运行如下程序:

```
MOV DL, 38H      ; 字符“8”的 ASCII 码送 DL
MOV AH, 2        ; 使用 DOS 的 2 号功能调用
INT 21H          ; 进入功能调用, 输出“8”
INT 20H          ; DOS 中断服务, 程序正常结束
```

该程序运行结果是在显示器上输出一个字符 “8”。如果要输出其他字符, 请改变程序中 “38H” 为相应字符的 ASCII 码。其中涉及 DOS 和 BIOS 功能调用。因为我们是在 DOS 的支持下运行汇编语言程序, 所以一般情况下, 不能轻易使用输入/输出指令直接通过端口输入或输出, 而必须使用 DOS 内部提供的子程序完成输入或输出。

DOS 功能调用就是为诸如此类的目的设置的。DOS 功能调用要求在进入 INT 21H 调用前, 首先将功能调用号送入 AH 寄存器, 并根据功能调用号准备初始数据。也就是说 INT 21H 的 2 号功能调用是输出 DL 寄存器中的字符。INT 20H 是 DOS 中断服务, 这一软中断用来正常结束程序。

运行步骤:

(1) 进入 DEBUG

假设 C 盘上有 DEBUG.COM 程序, 进入 DOS 环境后键入 DEBUG<ENTER>, 即:

```
C:\>DEBUG✓
```

随后屏幕显示:

“-”号是 DEBUG 的提示符,在该提示符下可键入任意 DEBUG 命令。现在用 A 命令输入程序。

(2) 输入程序并汇编

```
-A 100✓  
05AC:0100  MOV  DL, 38✓  
05AC:0102  MOV  AH, 2✓  
05AC:0104  INT  21H✓  
05AC:0106  INT  20H✓  
05AC:0108  ✓
```

至此程序已经输入完毕,并汇编成机器指令。接下来用 G 命令运行程序。

(3) 运行程序

```
-G=100✓  
8  
Program terminated normally
```

如果想分析该程序的指令,可使用反汇编命令 U 进行操作。

(4) 反汇编

```
-U 100, 108✓  
05AC:0100  B238          MOV  DL, 38  
05AC:0102  B402          MOV  AH, 02  
05AC:0104  CD21          INT  21  
05AC:0106  CD20          INT  20  
05AC:0108
```

(5) 将机器指令程序送到起始地址为 200H 的单元

```
-E 200 B2, 38, B4, 02, CD, 21, CD, 20✓
```

(6) 执行机器指令程序

```
-G=200✓  
8  
Program terminated normally
```

1.1.2 DEBUG 命令

1. DEBUG 命令概述

为便于学习,将 DEBUG 程序的主要命令列于表 1.1 中。在使用 DEBUG 命令时,

应注意以下一些 DEBUG 中的公共信息和操作技巧。

(1) 状态标志寄存器的初始状态为:

NV UP EI PL NZ NA PO NC

(2) 如果由 DEBUG 装入文件的扩展名为 .EXE, 则 DEBUG 必须重新定位和设置段寄存器, 堆栈指针和指令指针为该文件所定义的值, DS、ES 寄存器总是指向最低可用段的程序段前缀。

(3) 除命令 RF 外, DEBUG 命令本身均为单字符, 后面跟一个或多个参数, 有些命令不给参数也能执行, 如表中 [] 内的参数, 这种情况下采用默认值作为参数。

(4) DEBUG 下输入和显示的数均为十六进制数, 故数字后不可加后缀 H、O、B 等, 否则认为命令错。

(5) 命令和参数不区分大小写。

(6) 命令和参数间可用空格或逗号等分隔符分开, 但只有在两个相邻的十六进制数值之间的分隔符是必需的。

(7) 若命令输入有误时, DEBUG 会显示出错信息 “^ error”, 表示该命令使用失败。

(8) 有时可用 CTRL+BREAK 键停止命令的执行, 返回 DEBUG 提示符 “-” 状态。

(9) 若一个命令产生大量输出, 可用 CTRL+S 键暂停显示, 按任意键重新启动显示。

表 1.1 DEBUG 命令一览

分 类	命令格式	功能简介
读写寄存器	R	显示所有寄存器内容
	R 寄存器名	显示和修改指定寄存器内容
	RF	显示和修改标志寄存器内容
汇编和反汇编	A[内存地址]	从指定地址开始汇编指令
	U[内存块]	对指定内存块进行反汇编
执行指令	T[内存地址][条数]	单步或多步执行指令
	P[= 内存地址][条数]	单步或多步执行指令
	G[= 内存地址]	连续执行指令
	G[= 内存地址]断点地址	设断点执行程序
读写内存	D[内存块]	显示指定内存块内容
	E 内存地址 字符或数值串	修改指定内存内容
	F 内存块 字符或数值串	填充指定内存块
	S 内存块 字符串或数值	在指定内存块中查找串
	M 内存块 1 内存块 2 的首地址	复制内存块内容
	C 内存块 1 内存块 2 的首地址	比较两个指定内存块
读写磁盘	N[d:][PATH]文件名.扩展名	指定欲读写的磁盘文件
	W 内存地址	将指定内存块写入文件
	L[内存地址]	将文件调入内存
读写 I/O 端口	I 端口地址	读入指定端口的内容
	O 端口地址 数值	将数据写入指定端口
十六进制加减	H 数值 1 数值 2	计算并显示两数之和, 两数之差
退出 DEBUG	Q	退出 DEBUG, 返回 DOS

DEBUG 命令不同,其后面所需的参数也不同。下面简单说明参数的表示方法。

(1) 内存地址

内存地址表示一个内存区域的起始地址,由段地址和偏移地址两部分组成。段地址可用一个 1 至 4 位的十六进制数表示,也可用某个段寄存器表示,偏移地址只能用一个 1 至 4 位的十六进制数表示。段地址和偏移地址之间必须用冒号分隔。段地址部分可以缺省,此时段地址由 DEBUG 程序的默认关系确定,默认关系的规律是:各命令沿用上一次该命令中使用的段值;当命令首次使用而命令中未给出段值时,则命令 A, U, I, T, P, G 默认的段值为当前 CS 的值,而 D, E, F, M, C, W, S 等命令默认 DS 的值为段值。如:

300:8000	表示 300:800H 的内存地址;
CS:100	表示以 CS 的内容为段地址,偏移地址为 100H 的内存地址;
100	表示段地址为默认值,偏移地址为 100H 的内存地址;

(2) 内存块

内存块表示一个内存区域,有两种表示方法;一种是用首地址和末地址表示,其中首地址用前面介绍的内存地址表示法,而末地址只用偏移地址表示,其段地址默认为首地址的段值,末地址的偏移地址应比首地址的偏移地址大。另一种是用首地址和长度表示,其中长度由字符 L 引出。内存块的最大值是 64K。有些命令要求给出的是内存块,而命令只给出一个首地址时,则按默认长度完成命令功能,如命令 D 默认的长度为 80H 个字节,命令 U 默认的长度为 20H 个字节。

如下面两种表示法均表示 DS:0200 ~ DS:024FH 的内存块:

```
DS:200 24F
DS:200 L50
```

如下面两种表示法均表示 3000H:5100 ~ 3000H:51FFH 的内存区域:

```
3000:5100 51FF
3000:5100 L100
```

(3) 字符串参数可用双引号或单引号给出。如:

```
-F DS:1000 'Please input:' ✓
```

(4) 数值参数为一个字节的十六进制数。多个数值参数之间必须用分隔符分隔。
如:

```
-E DS:2000 12 34 56 78 90 ✓
```

(5) 端口地址的范围为 0 ~ 0FFFFH。

2. 常用 DEBUG 命令介绍

下面是按功能分类的常用 DEBUG 命令的使用说明。

(1) 显示、修改寄存器内容