

21世纪高职高专信息类专业系列教材

IBM-PC汇编语言实用教程

主 编 程时兴
副主编 贾建中
易海胜
主 审 苏光奎



重庆大学出版社

21 世纪高职高专信息类专业系列教材

IBM-PC 汇编语言实用教程

主 编 程时兴
副主编 贾建中 易海胜
参 编 张晓波 陈伟芳
主 审 苏光奎

重庆大学出版社

· 内 容 提 要 ·

本书共分九章,基础知识、寻址方式与指令系统、汇编语言程序格式与上机过程、程序设计基本技术、子程序、程序设计的其他技巧和方法、输入/输出与常用中断服务程序、磁盘文件创建和访问、汇编与C语言混合编程。介绍IBM-PC汇编语言的基本知识和程序设计方法,汇编语言与C语言混合编程技巧,汇编语言开发程序具体操作步骤,并从应用的角度说明汇编语言在实际工作中的具体应用。

本教材参考教学时数为60学时。

本教材强调基础并注重与实际应用相结合,内容深入浅出;主线明确。推荐作为高职高专相应课程的教材,也供使用汇编语言的工程技术人员参考。

图书在版编目(CIP)数据

IBM-PC 汇编语言实用教程/程时兴主编. —重庆:重庆大学出版社,2002.2

21世纪高职高专信息类专业系列教材

ISBN 7-5624-2172-2

I.I... II.程... 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(2001)第094394号

·21世纪高职高专信息类专业系列教材·

IBM-PC 汇编语言实用教程

主 编 程时兴

副 主 编 贾建中 易海胜

主 审 苏光奎

责任编辑 肖顺杰

*

重庆大学出版社出版发行

新华书店经销

重庆科情印务有限公司印刷

*

开本:787×960 1/16 印张:17.75 字数:338千

2002年2月第1版 2002年2月第1次印刷

印数:1~5 000

ISBN 7-5624-2172-2/TP·255 定价:23.00元

· 系列教材编委会 ·

主任单位：

重庆电子职业技术学院

副主任单位：

武汉职业技术学院

邢台职业技术学院

陕西工业职业技术学院

贵州大学职业技术学院

编委(以姓氏笔画为序)：

才大颖

刘真祥

李传义

张 洪

李永平

涂湘循

程迪祥

王晓敏

刘业厚

吕何新

张中洲

杨滨生

唐德洲

樊流梧

王兆其

刘建华

张学礼

张国勋

林训超

徐民鹰

黎省三

王柏林

朱新才

张明清

张西怀

赵月望

曹建林

· 系列教材参编学校(排名不分先后) ·

武汉职业技术学院
重庆电子职业技术学院
陕西工业职业技术学院
邢台职业技术学院
贵州大学职业技术学院
河南职业技术学院
三门峡职业技术学院
湖南工业职业技术学院
昆明大学
广西机电职业技术学院
成都电子机械高等专科学校
昆明冶金高等专科学校
珠海职业培训学院
广东交通职业技术学院
浙江省树人大学
江西工业职业技术学院
成都航空职业技术学院
辽宁仪器仪表工业学校
北京信息职业技术学院
徐州交通职业技术学院
重庆大学职业技术学院
重庆邮电学院
重庆工业高等专科学校
重庆石油高等专科学校
重庆职工大学
西南农业大学
长沙航空职业技术学院
番禺职业技术学院

总 序

当今世界,科学技术的发展日新月异。在这空前的技术发展进程中,电子信息技术以其独特的渗透力和亲和力,正在迅速地改变着我们周围的一切。利用现代电子信息技术来改变我们的生活与学习,改造传统的各行各业,已成为当今社会人们的共识。

教育在我国社会主义建设发展进程中所具有的战略地位和基础作用已被越来越多的人所认识。职业技术教育、特别是高等职业技术教育在近二十年来得到了长足的发展,“高等教育法”、“职业教育法”的颁布与实施,使我国高等职业教育步入了法制轨道,国家与社会的进步与发展,需要高等职业教育,技术的进步与发展,也需要高等职业教育,高等职业教育成为世界教育发展的共同趋势。

在我国国内,高等职业教育毕竟是一种新型的教育类型,发展历史不太长,在教育观念、教育体制、教育结构、人才培养模式、教育内容、教学方法、教材、教法诸方面,有不少问题需要研究与探索。重庆大学出版社从促进高等职业教育发展战略的角度,于1999年邀请国内三十余所长期开办电子信息类专业的学校,开展对电子信息类高职、高专教材的开发研讨。与会学校有独立设置的职业技术学院、高等专科学校、职业大学、普通高校中的职业技术学院、多年试办高职班的重点中专学校。大家一致认为,我国高等职业教育的教材建设非常薄弱,基本上没有自己的教材,从而导致针对性、适应性差。从电子信息类专业角度看,缺乏成体系的系统教材,从而导致不同层次教材的交叉重复现象严重;再者,现行教材中缺乏对新技术、新工艺、新产品相关内容的介绍。因此,开发适应新世纪高等职业技术教育的教材就成为当务之急,它总的原则应是:根据培养应用型、技能型人才目标,从职业岗位对专业知识的需要来确定教材的知识深度及范围,坚持“必须、够用”的原则;同时注意知识的应用价值在教材中的科学体现,力求构筑具有高职特色的理论知识体系;基本概念、基本原理以讲明为度,同时将一些内容相近的部分进行合并。另外,针对高

职业教育培养技能型、现场型人才的目标,把训练职业能力的实践技术体系方面的内容,与理论知识体系有机地结合起来,力求在这方面采用先解决有无问题,再解决提高与系统性问题的原则,我们在一开始就力求站在一个较高起点上,先从电子信息类教材开发做起,然后再进一步开发其他专业大类的应用型高职教材。

经过近一年的努力,电子信息类高职、高专系列教材就要与大家见面了。本系列教材的编写原则、编写体例均是根据教育部高职高专培养目标并由参与系列教材编写的全国三十余所相关院校经过数次研讨、反复论证确定的。尽管我们对它报有较高的期望,但这毕竟是一个新生事物,是一种尝试,成功与否,这条路我们会一直走下去。为了我们共同的高职教育事业,欢迎大家在使用过程中,指出它的不足,以利于我们今后的工作。

编委会

2000年7月

——前言——

我们知道计算机中能执行的程序是机器语言程序,而汇编语言的执行指令就是机器语言指令的符号化。汇编语言是计算机中运行最快、占用空间最小的编程语言,也是能方便、直接控制计算机硬件的语言。

学习汇编语言的意义:

- 了解计算机的工作过程。从打开计算机电源的那一刻开始,计算机就是在执行程序,直至关闭计算机,执行的程序和处理的数据必须先取到内存中才能运行和处理。

- 直接对外部设备进行I/O控制。
- 编写要求占用空间小,或执行速度快的程序。
- 通过一些软件工具,分析现有的程序或计算机病毒程序。
- 编写没有操作系统环境支持的控制程序。

本书以8086/8088 系统为背景,强调基础、突出应用,本着理论够用为度,理论为实践服务,帮助学生学以致用原则,讲述8086/8088 CPU 和内存的结构、寻址方式、指令系统、汇编语言程序设计的常用方法和技巧、汇编与C语言混合编程、DEBUG调试程序的应用。以具体应用事例为导向贯穿全书,阐明汇编语言在键盘输入、显示控制、CMOS信息读写、软驱和硬盘的直接访问控制、磁盘文件操作等方面的应用。

Intel286、386、486、Pentium及以后PC系列CPU都完全包含8086/8088指令集,也就是说,在这些CPU构成的计算机中可以直接运行8086/8088中编写的程序。

本书编写的两点说明:

(1)汇编语言本身对字母大小写不敏感,为了便于教与学,凡用户定义的变量名、过程名、标号名等使用小写,关键字等使用大写形式。

(2)在文字叙述中(不是在程序中),寄存器内容通常采用加括号的方式表示,如 $AX \leftarrow (AX) + 1$,但本书表示寄存器内容不加括号,如 $AX \leftarrow AX + 1$ 。这只是描述的习惯,作者认为这样不会有歧义,而且这样描述更明确、简单。

从理论上讲,利用汇编语言可以实现对计算机的所有控制,完成任何复杂任务的处理,但事实上,对于复杂任务使用汇编语言是不合算的,而利用高级语言编程或汇编语言与高级语言(如C语言)混合编程,能起到事半功倍的效果。所以根据任

务的性质,选择适当的程序设计语言环境,是非常重要的,也决定了程序设计者完成任务的质量和效率。

本书由武汉职业技术学院程时兴任主编,邢台职业技术学院贾建中、武汉职业技术学院易海胜任副主编,河南职业技术学院张晓波、浙江树人大学陈伟芳任参编。程时兴编写第七章、第八章、第九章和第三章第六节;贾建中编写第四章和第三章第一节至第五节;易海胜编写第一章、第六章;张晓波编写第五章、附录;陈伟芳编写第二章。全书由程时兴统稿。

武汉大学苏光奎教授在百忙中审阅了全书,提出了许多宝贵意见,在这里表示真诚的感谢,同时感谢重庆大学出版社的通力合作。本书力争写出编者在教学过程中积累的经验和体会,但疏漏之处在所难免,真诚希望读者批评指正。

作者

2001年9月

目录

1	第一章 基础知识
1	第一节 汇编语言及其应用
3	第二节 进制及进制转换
8	第三节 计算机中的数据表示
12	第四节 IBM-PC微型计算机基本结构
13	第五节 8086/8088 CPU结构及其寄存器
18	第六节 内存和内存地址
22	第七节 堆栈
25	第八节 Intel X86系列CPU软件兼容性
26	小结
26	思考题
28	第二章 寻址方式与指令系统
28	第一节 寻址方式
32	第二节 指令系统
59	第三节 指令编码
62	小结
62	思考题
66	第三章 汇编语言程序格式与上机过程
66	第一节 汇编语言中的语句格式
68	第二节 汇编语言程序中的数据
74	第三节 汇编语言程序格式
78	第四节 其他常用伪指令
80	第五节 汇编语言程序上机过程
82	第六节 DEBUG 调试程序

92	小结
93	思考题
95	上机题
96	第四章 程序设计基本技术
96	第一节 程序设计概述
101	第二节 顺序程序设计
104	第三节 分支程序设计
114	第四节 循环程序设计
128	第五节 程序设计综合举例
131	小结
132	思考题
134	上机题
137	第五章 子程序
137	第一节 子程序的格式
141	第二节 子程序的调用与返回
147	第三节 主程序与子程序的参数传递
159	第四节 子程序嵌套调用
163	*第五节 子程序递归调用
166	小结
167	思考题
167	上机题
169	第六章 程序设计的其他技巧和方法
169	第一节 块操作
175	第二节 宏功能程序设计
181	第三节 程序模块的连接
183	小结
184	思考题
186	上机题

187	第七章 输入/输出与常用中断服务程序
187	第一节 输入/输出(I/O)指令
190	第二节 访问CMOS信息
193	第三节 中断与中断处理程序
200	第四节 中断服务程序调用与返回
201	第五节 键盘和显示控制
213	第六节 软盘和硬盘访问控制
221	第七节 控制扬声器
225	小结
226	思考题
226	上机题
228	第八章 磁盘文件创建和访问
228	第一节 DOS 功能调用INT 21H 的文件功能
233	第二节 创建文件和访问文件的基本操作过程
237	第三节 磁盘文件操作举例
240	小结
240	思考题
241	上机题
242	第九章 汇编和C语言混合编程
242	第一节 C语言程序调用汇编语言过程
247	第二节 Turbo C行间汇编
249	第三节 混合编程举例
251	小结
252	思考题
252	上机题
253	附录
253	附录一 IBM-PC键盘ASCII码和键盘扫描码
255	附录二 汇编程序出错信息
258	附录三 8088指令表
265	附录四 DEBUG 命令一览表

267	附录五 硬盘分区表存放格式
268	附录六 CMOS 标准信息寄存器地址分配
270	参考文献

第一章

基础知识

本章要点

- 机器语言 汇编语言
- 二进制、十六进制及相互转换
- 数据在计算机内的存储 汉字编码
- 微机的基本结构
- 8086/8088 CPU 寄存器
- 内存 物理地址 逻辑地址
- 堆栈

本章介绍了计算机中特殊用到的二进制数和十六进制数，讲述了数据在计算机内的存放形式。并且从微型计算机的基本结构入手，着重介绍了 8086/8088 CPU 的寄存器组和内存的组织形式，是学习本课程的基础知识。

第一节 汇编语言及其应用

一、机器语言

我们知道计算机是通过逐条地执行组成程序的指令来完成人们所给予的任务的，所以计算机只能识别并直接执行用二进制代码编写的语句，由二进制代码组成的指令称为机器指令，机器指令的集合称为机器语言。

很显然机器语言对于人们是很不方便的。用这种机器语言制作复杂的程序可以说是一件非常困难的事情，只有专门的技术人员才能进行，所以，局限性很大。在计算机发明的初期，除了用机器语言制作程序外，别无他法。随着计算机应用范围的扩大，程序越来越大，越来越复杂，用机器语言来制作程序是越来越困难了。于是

人们开始寻求尽可能简单的制作程序的各种方法。

二、汇编语言

计算机能识别的惟一语言是机器语言,而这种语言编写程序很不方便,所以在计算机语言的发展过程中就出现了汇编语言和高级语言。汇编语言是一种符号语言,它几乎和机器语言一一对应,但在书写时却使用由字符串组成的助记符。例如:加法在汇编语言中是用助记符 ADD 表示的,而机器语言则用 6 位二进制代码来表示。显然,相对于机器语言来说,汇编语言是易于理解的,但计算机却不能直接识别汇编语言。汇编程序就是用来把由用户编制的汇编语言程序翻译成机器语言程序的一种系统程序。IBM-PC 机中的汇编程序有 ASM 和 MASM 两种。

三、汇编语言的应用

凡是学过一种程序设计的高级语言,都会有高级语言“易学好用”的感觉,这是因为这些语言的语句是面向数学语言或自然语言的,因此容易接受、掌握。相对来说汇编语言编制程序比用高级语言要困难些。既然如此,为什么还要学习和使用汇编语言呢?

(1)学习和使用汇编语言可以从根本上认识、理解计算机的工作过程。因为一台计算机执行一个任务,归根到底就是执行一个计算机语言程序。通过用汇编语言编制程序,可以更清楚地了解计算机是怎样完成各种复杂的工作。在此基础上,程序设计人员更能充分地利用机器硬件的全部功能,发挥机器长处。这就如同你编制的程序在计算机上运行时,它调动并控制着机器中每个部件和电路。

(2)现在计算机系统中,某些功能仍然靠汇编语言程序来实现。例如机器的自检,系统的初始化,实际的输入输出设备的操作,至今仍然是用汇编语言编制的程序来完成的。

(3)汇编语言程序的效率通常高于高级语言程序。这里的“效率”是指程序的目标代码的长短和程序运行的速度。所以在节省内存空间和提高运行速度是重要指标的场合,如实时过程控制,常常是用汇编语言来编制程序。

鉴于以上理由,现在许多高级语言都设置有与汇编语言程序接口的功能,便于用户用汇编语言编制某些子程序,完成与机器联系紧密的特定功能,提高高级语言程序的效率。

第二节 进制及进制转换

一个数值型的数据可以采用不同的进制表示,其值大小相同。人们习惯的是十进制,计算机中采用的是二进制,为了便于书写和阅读,又必须了解八进制和十六进制。

一、十进制

进制即进位计数制,是一种计数的方法。习惯上最常用的是十进制计数法。一个任意的十进制数可表示为:

$$a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$$

其含义是:

$$a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_0 \times 10^0 + b_1 \times 10^{-1} + b_2 \times 10^{-2} + \cdots + b_m \times 10^{-m}$$

其中 $a_i (i=0, 1, \cdots, n)$, $b_j (j=1, 2, \cdots, m)$ 是0, 1, 2, 3, 4, 5, 6, 7, 8, 9十个数码中的一个。

十进制数的基数为10,即数码的个数为10,且遵循逢十进一的规则。上式中相应于每位数字的 10^k 称为该位数字的权,所以每位数字乘以其权所得到的乘积之和即为所表示数的值。例如:

$$12345.67 = 1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 7 \times 10^{-2}$$

十进制数是人们最熟悉、最常用的一种数制,但它不是惟一的数制。例如计时的时、分、秒就是按60进制计数的。基数为 r 的 r 进制数的值可以表示为:

$$a_n \times r^n + a_{n-1} \times r^{n-1} + \cdots + a_0 \times r^0 + b_1 \times r^{-1} + b_2 \times r^{-2} + \cdots + b_m \times r^{-m}$$

其中 a_i, b_j 可以是0, 1, $\cdots, r-1$ 中的任一个数码, r^k 则为各位数相应的权。

二、二进制

计算机中为方便存储及计算的实现,采用了二进制数。二进制数的基数为2,只有0, 1两个数码,并遵循逢二进一的规则,它的各位权是以 2^k 表示的,因此二进制数 $a_n a_{n-1} \cdots a_0 b_1 b_2 \cdots b_m$ 的值是:

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots + b_m \times 2^{-m}$$

其中 a_i, b_j 为0, 1两个数码中的一个。例如:

$$101101_2 = 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 = 45_{10}$$

其中数的下标表示该数的基数 r ,即二进制的101101与十进制的45等值。

n 位二进制数可以表示 2^n 个数。例如3位二进制数可以表示8个数,见表1-1。而4位二进制数则表示十进制的0~15共16个数,见表1-2。

表1-1 3位二进制数表示的8个数

二 进 制 数	000	001	010	011	100	101	110	111
相应的十进制数	0	1	2	3	4	5	6	7

表 1-2 4 位二进制数则表示的 16 个数

二 进 制 数	0000	0001	0010	0011	0100	0101	0110	0111
相应的十进制数	0	1	2	3	4	5	6	7
二 进 制 数	1000	1001	1010	1011	1100	1101	1110	1111
相应的十进制数	8	9	10	11	12	13	14	15

为便于人们阅读及书写,经常使用八进制数或十六进制数来表示二进制数。它们的基数和数码表示如表1-3所示。

表1-3 几种常用的进位计数制的基数和数码

进 位 计 数 制	基 数	数 码
十六进制数	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
十进制数	10	0,1,2,3,4,5,6,7,8,9
八进制数	8	0,1,2,3,4,5,6,7
二进制数	2	0,1

按同样的方法,读者可以很容易地掌握八进制和十六进制数的表示方法。可以看出: 23_{10} 可以表示为 17_{16} , 27_8 及 10111_2 ; 1.375_{10} 可以表示为 1.6_{16} , 1.3_8 及 1.011_2 等。在计算机里,通常用数字后面跟一个英文字母来表示该数的数制。十进制数一般用D(Decimal),二进制数用B(Binary),八进制数用O(Octal),十六进制数用H(Hexadecimal)来表示。例如:117D,1110101B,0075H,……当然也可以用这些字母的小写形式。

三、十六进制

在计算机内部,数的运算和存储都是采用二进制的。但是,二进制对于人的阅读、书写及记忆都是很很不方便的。十进制数虽然是人们最熟悉的一种进位计数制,但它与二进制数之间并无直接的对应关系。为了便于人们对二进制数的描述,应该选择一种易于与二进制数相互转换的数制。显然,使用 2^n 作为基数的数制是能适合人们的这种要求的,常用的有八进制数和十六进制数,这里主要介绍十六进制数。

计算机中存储信息的基本单位为一个二进制位(bit),它可以用0和1两个数码来表示一个数。此外,由于计算机中常用的字符是采用由8位二进制数组成的一个