IBM-PC宏汇编语言

程序设计

中国科学院　尹彦芝

水利电力出版社

IBM-PC宏汇编语言程序设计

中国科学院　尹彦芝

水利电力出版社

IBM-PC宏汇编语言程序设计

中国科学院　尹彦芝

*

水利电力出版社出版

（北京三里河路6号）

新华书店北京发行所发行·各地新华书店经售

水利电力出版社印刷厂印刷

*

787×1092毫米　32开本　12印张　261千字

1987年8月第一版　1987年8月北京第一次印刷

印数00001—10860册　定价：2.50元

书号　15143·6413

内 容 简 介

本书系统深入地阐述了IBM-PC机的宏汇编语言的程序设计方法。全书共九章，前三章从实用的角度介绍8086/8088的寻址方式和指令系统。第四章用大量实例，详细讲解了IBM-PC宏汇编语言所有的伪指令和操作符。第五、六章介绍宏汇编语言与操作系统的关系。第七章介绍ROM BIOS的使用办法。第八、九章面向较高水平的程序员，介绍程序常驻内存的技巧及高级语言与宏汇编语言的连接办法。

本书力求概念清楚，所附大量典型程序均完整，可上机检查。该书适用于在IBM-PC上从事硬件和软件开发的人员使用，也可供大专院校有关专业师生参考。

前　言

目前，微型计算机已应用于科研、生产、管理等各个领域。在当今众多的微型计算机中，IBM公司的微型计算机占据着最重要的位置。1981年，IBM公司推出了IBM-PC机。1983年，IBM公司又推出了IBM-PC/XT机。这两种机器在微型机的销售市场上占有绝对优势，再加上种类繁多的IBM-PC和IBM-PC/XT的兼容机，构成了当今微型计算机市场上最重要的一簇。这些机器使用的中央处理器都是8088和8086，其汇编语言的使用大致是相同的，本书所讲述的虽然主要是PC-DOS支持下的IBM-PC和IBM-PC/XT机的宏汇编语言，但对这一大簇微型计算机上使用的汇编语言都有参考价值。

学习汇编语言的原因基本上有两个。第一，汇编语言具有目标码短、执行速度快、能够充分利用硬件资源及能够进行精确控制的优点，这些优点是其它高级语言远远不能比拟的。这也正是在某些场合下（例如，自动控制和在开发系统软件时），应该使用甚至必须使用汇编语言的原因。人们对8086/8088越熟悉，使用这种汇编语言来构成小型控制系统或使用含这种汇编语言的单板机的机会就会越多，此时就越感到汇编语言的重要了。第二，汇编语言是一种低级语言，它与硬件的关系最密切。常常看到这样的现象，有些人已经学会了BASIC或FORTRAN之类的高级语言，但对计算机究竟是怎样工作的却知之不多，因而影响了对计算机的全面掌握和

使用。要了解计算机内部工作的奥秘，充分利用硬件资源、发挥机器的特点，就必须学习汇编语言。

正因为汇编语言是面向机器的语言，需要有更多的有关计算机硬件和计算机操作系统方面的知识，故较BASIC之类的高级语言难于掌握。与一般汇编语言相比，IBM-PC机宏汇编语言的难点主要体现在两个方面。一个是程序分段的概念，另一个是变量和过程类型的概念。另外，还增加了一般汇编语言所不具有的数据结构。

本书面向已具备计算机基础知识，并多少使用过计算机的读者，在简单介绍基础知识之后，对IBM-PC宏汇编语言程序设计作较全面系统地介绍。本书力求内容深入而实用，使在IBM-PC机上从事软件和硬开发的同志能以它作为实际工作的指南和手册。

本书经北京计算机学院讲师张怀莲同志审稿；全书的撰写得到了中国科学院孙凤霞同志的大力帮助，在此一并致谢。敬请读者对本书的缺点错误提出宝贵意见。

作　者

目　录

前　言
第一章　几个基本概念 …… 1
第一节　数制和编码 …… 1
第二节　数据类型 …… 3
第三节　数的存贮 …… 6
第四节　子程序和堆栈 …… 10
第五节　中断 …… 14
第二章　寻址方式 …… 22
第一节　8086/8088的寄存器 …… 22
第二节　有效地址的计算 …… 29
第三章　指令系统 …… 35
第一节　数据传送指令 …… 35
第二节　算术指令 …… 44
第三节　位操作指令 …… 51
第四节　字符串指令 …… 54
第五节　程序转移指令 …… 61
第六节　处理机控制指令 …… 67
第四章　伪指令和伪操作符 …… 70
第一节　宏汇编语言的基本概念 …… 70
第二节　数和变量 …… 78
第三节　程序结构 …… 90
第四节　标号和过程 …… 100
第五节　结构 …… 111
第六节　记录 …… 116

第七节 表达式 ……122
第八节 多模块的连接 ……127
第九节 段组 ……133
第十节 宏指令 ……140
第十一节 条件汇编 ……157
第十二节 其他伪指令 ……169
第五章 系统调用……171
第一节 系统调用概述 ……171
第二节 字符设备系统调用 ……173
第三节 文件管理系统调用 ……178
一、文件控制块(FCB)的概念 ……178
二、一个应用文件管理系统调用的例子 ……182
三、一个应用结构和文件管理系统调用的例子 ……192
四、采用文件控制字进行文件管理的例子 ……194
第六章 汇编语言的上机过程 ……201
第一节 行编辑程序EDLIN ……202
第二节 宏汇编程序MASM ……205
第三节 连接程序LINK……207
第四节 执行程序 ……209
第五节 调试程序DEBUG ……212
第六节 ·COM文件和·EXE文件……215
第七章 ROM BIOS的使用 ……222
第一节 ROM BIOS的程序清单 ……223
第二节 上电自检 ……225
第三节 ROM BIOS中断向量 ……226
第四节 设备驱动程序详解 ……231
一、系统服务程序……231
二、打印机驱动程序……234
三、异步通讯驱动程序……239

四、键盘驱动程序……242
五、软盘驱动程序……243
六、显示驱动程序……255
七、硬盘驱动程序……271
第八章　程序常驻内存的办法……279
第一节　概述……279
第二节　低地址法……280
第三节　高地址法……292
第九章　汇编语言与高级语言的连接……306
第一节　概述……306
第二节　BASIC语言与汇编语言的连接……307
一、BASIC语言与汇编语言连接……307
二、用BLOAD命令装入的例子……314
三、用赋值语句装入的例子……322
第三节　FORTRAN语言与汇编子程序的连接……326
附录一　指令系统一览表……329
附录二　PC DOS2.0版的系统调用……348

第一章 几个基本概念

第一节 数制和编码

一、二进制

在所有电子数字计算机内部，数据基本上都是用二进制来表示的。二进制数与十进制数相比，有两点明显的不同：

（1）10进制数有10个不同的数字符号，即0,1,2…9。二进制数只有两个数字符号：0和1。

（2）十进制数是逢10进1的，而二进制数是逢2进1的。例如，二进制数101101所表示的数可以用数学式子表示为：

$$101101B=1\times2^5+0\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0$$
$$=32+0+8+4+0+1$$
$$=45$$

为了与十进制数相区别，我们在这里用了后缀符号"B"。

二、补码

日常生活中离不了负数的概念，计算机也必须引进这个概念，也就是说，计算机应该能表达带符号的数。在8088／8086中是采用二的补码来表示带符号数的。在这种表示法中，数的最左边的一位是用来表示符号的，如果是正数，这一位是0，如果是负数，这一位是1。对于正数，用补码表示后的结果与相应的不带符号数的表示结果将是完全一样的。对于负数，情况就不一样了，为了求得它的补码，应该先把相应的无符号数的各位求反（由0变1，由1变0），再把求

反的结果加1。以4位二进制数为例，无符号数5应为0101B，+5的补码也应为0101B，－5的补码则应为1011B。

一个二进制数如果具有n位，可以编码出2^n种情况，可以代表2^n个不同的数。在补码制下，在这2^n个不同的数中，有（$2^{n-1}-1$）个用来表示正数，1个用来表示0，2^{n-1}个用来表示负数，也就是说能够表达的数的范围是从（-2^{n-1}）到＋（$2^{n-1}-1$）。对于8位二进制数，能够表达的数的范围是从－128到＋127。对于16位二进制数，能够表达的数的范围是从－32768到＋32767。

三、16进制

前面已经谈到过，计算机本质上只能处理0和1这样两个数字，所以计算机最易采用二进制数。但是，二进制数写起来特别麻烦，容易出错。为了书写起来紧凑一些，人们又采取了16进制数。

与上面讨论的十进制数和二进制数相比较，16进制数具有如下特点：

（1）16进制数有16个不同的数字符号，它们是0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。

（2）16进制数是逢16进1的，其后缀符号为H。

因为在16进制数数字字符中，除了正常的10个数字外，还增加了A、B、C、D、E、F这6个字母，这就会引起新的问题。例如字符串“FAH”既可以被看是一个16进制的数“FA”，又可以被看作是一个变量名，从字面上无法加以区别。为了能够把这两者加以辨认，在汇编语言程序中必须遵循一条原则，数必须以0到9这10个数字开头，变量名必须以一个非数字字母开头。根据这个原则，字符串“FAH”应该被看作是变量名而不是16进制数。如果一定要表示“FA”

这个16进制数，则应写成“OFAH”。

第二节 数 据 类 型

数据类型就是存贮于计算机内表示数据的格式或编码方案。在IBM-PC机中的数据类型可有如下几种：逻辑值、无符号整数（序数）、有符号整数（简称整数）、浮点数、二——十进制数（简称BCD码）、字符、字符串及地址指针。

一、逻辑值

逻辑值只有两个可能值，用于表示只具有两个状态的量。日常生活中碰到的真与假、开与关、对与错、是与非等都可以用逻辑值来表示。在机器中，逻辑值也还常用于这样的目的：作为状态标志，表示某设备已经准备好输入或输出；作为条件标志，程序执行时，帮助判断是否应该转移。

二、序数

序数是不带符号的整数，它是从0开始计数的。很显然，在机器中用来表示序数的二进制数的位数越多，所能表示的序数的范围也就越大。二进制数为8位时，可以表示的序数值是从0到255；二进制数为16位时，可以表示的序数的值是0到65535。

三、整数

整数也叫做带符号整数。在所有的微处理器中，整数都是以二进制补码表示。8位时，可以表示的整数值是从－128到＋127。16位时，可以表示的整数值是从－32768到＋32767。

四、浮点数

数学中，科学记数法表示的数是由符号、指数和尾数组成的。例如：

数	科学记数法	符号	尾数	指数
127	1.27×10^{2}	+	1.27	2
−0.67	-6.7×10^{-1}	−	6.7	−1
6925	6.925×10^{3}	+	6.925	3

在计算机中，为了表示一个浮点数，通常用如下三种格式：32位、64位和80位。头两种格式(32位和64位)分别叫做短实数和长实数，是IEEE浮点数标准推荐的；第三种格式用在8087数值数据处理器内部，叫做INTEL暂用实数格式。

在计算机中引进浮点数以后，就出现了如何表示小数点及小数点在数中位置的新问题。计算机没有一个方便的办法表示它们，故采取了隐含的法则，这就是认为尾数中的整数位只有一位，小数点紧跟在这个整数的后面(即浮点表示法)，如图1-1所示（以64位为例）。

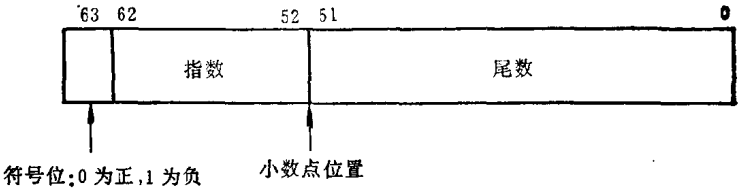

图 1-1

在浮点数中还有一个问题值得注意，这就是规格化的问题。所谓规格化就是尾数的最前一位(也就是尾数的整数位)应该一定是1（在二进制中），而不应该是0，使得尾数能表示的有效位数最多。如果实际的尾数不是这样，则尾数应该进行相应的左移或右移（指数也应相应的减去或加上这个移位的次数），以便符合规格化规则。既然尾数的整数位永远为1，为了节省内存空间，在计算机中也就根本不存贮这

一位。所以，实际上尾数总是在1和2之间。

数的实际小数点位置是由指数来决定的。正如上面的例中所示，指数本身也是有正有负的。与尾数不一样的是，指数不是采取补码表示法，而是采取偏码表示法。指数的偏码是用实际的指数再加上一个常数构成的，从而使得偏码指数恒为正。在64位浮点数表示法中，这个常数是1023(3FFH)。例如，如果实际指数是－3，则偏码指数将是1020。例如：

浮点数	二进制表示法
10	4024000000000000
100	4059000000000000
－127	C05FC00000000000

短实数格式的32位包含一位符号位、8位指数位、23位尾数位，实际上尾数有24位，其中的第1位恒为1，已不存贮。最小可能正数是2^{-126}，约等于1.175×10^{-38}；最大可能的正数是2^{128}，约等于3.40×10^{38}。

长实数格式的64位包含1位符号位、11位指数位、52位尾数位（实际上尾数有53位，其中的第1位恒为1，已不存贮）。最小可能的正数是2^{-1022}，约等于2.23×10^{-308}，最大可能的正数为2^{1024}，约等于1×10^{308}。

在INTEL暂用实数格式的80位中，包括1位符号位、15位指数位、64位尾数位。最小可能的正数为2^{-16382}，约等于3.36×10^{-4932}，最大可能的正数为2^{16382}，约等于1.19×10^{4932}。这个格式与其它格式不同之点在于所有尾数实际上都存贮了。在汇编语言中引进浮点数只是为了供8087处理器使用。

五、二——十进制编码

二——十进制编码是用4位二进制数来代表一个10进制

数字，这对于商业应用是很有利的，8086/8088CPU有特殊的指令来对这种类型的数进行处理。

六、字符

字符是用来表示字母数字和其它一些符号的。对于一些基本符号，几乎所有的机器的编码都是一样的，都是采用ASCII码。在ASCII码中，每一个字符指定一个独有的数。例如，A用10进制65表示，B用10进制66表示等。

七、字符串

字符串是由一串字符组成的，可用作记载逐字逐句的信息，例如提示信息，错误信息等。在某些文本编辑程序中，把整个文本文件都看作是一个字符串，此时字符串可能有成千上万个字符。既然字符串的长度是可变的，就一定需要有某种办法来表示这个长度，一般采取如下两种办法。一种办法是在字符串的开头（或其它地方）专门开辟一个或两个字节来指示字符串的长度，另一种办法是在字符串的结尾以一个特殊的字符（例如字符“$”）来表示字符串的终止。

八、指针

指针是用来指示地址用的，也就是说，它的内容是一个地址。在8088/8086微处理器中，一个物理地址是以两个16位数来存放的，低16位数用来存放偏移量，高16位数用来存放段地址，这两个16位数按特殊方法进行联合，构成20位的实际地址。关于偏移量和段地址，以后还要反复讲到，这里暂不多述。

第三节 数的存贮

数的存贮讲的是数在内存中如何存放。上面已经讲到，

在内存中最小可寻址的单元是字节。但是，在指令和程序中，可寻址的单元可以小到半字节和位，也可以大到字节、字、双字、四字、80位字和块，如图1-2所示：

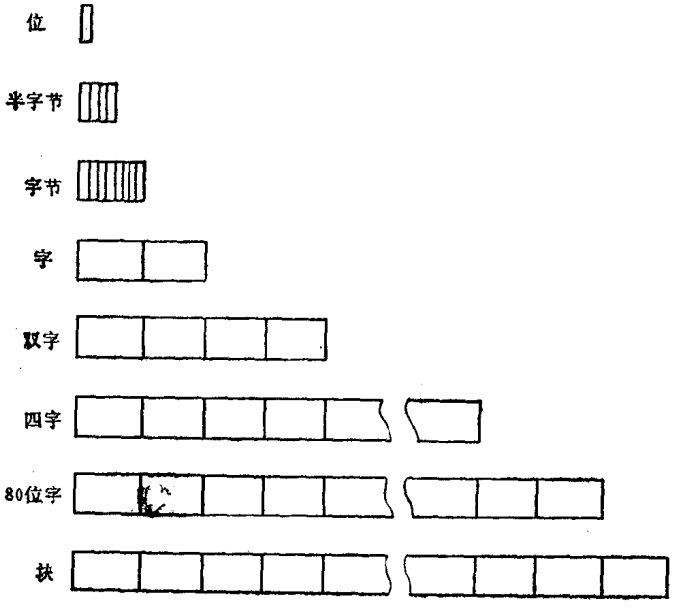

图 1-2

一、位

位是计算机存贮的最小的单元，它作为一个两状态的信号被存贮和传递。位可以用来存贮逻辑值或特征值。

二、半字节

半字节含有 4 位，它适合于用来存放一个二——十进制码（BCD码）或一个16进制数字。

三、字节

一个字节含有 8 位，等于两个“半字节”。字节适合于

用来存放一个字符（ASCII码），一个从0到255的序数，一个从－128到＋127的整数，两个BCD数（0到99）。有时候也用一个字节来存放一个逻辑值或一个BCD数。

四、字

这里说的字不是一般意义下的字，而是代表计算机中可以一次进行处理的最大二进制位数。既然在各种计算机中可以一次进行处理的最大二进制位数不同，那么各种计算机的字的含义也就不同。对于IBM系统370机，字是32位；对于IBM-PC机，字是16位。顾名思义，双字就是两个字，四字就是四个字，在IBM-PC机中分别为32位和64位。字，双字，四字和80位字常用来存放浮点数。

既然字是由两个字节组成的，就存在着这两个字节在内存中存放的先后次序问题。

INTEL公司是按如下原则来处理这个问题的：低地址存放低字节，高地址存放高字节。应该注意的是在四字的存贮问题上，MOTOROLA公司或ZILOG公司与INTEL公司的处理原则是不一样的。图1-3示出了两者的不同。

与其它一些计算机不同，8086/8088不要求字一定从偶地址开始存放，尽管在8086中从偶地址开始存放会加快一些存取的速度（此时叫做字是对准的）。字节、字、双字、四字以及指令可以任意混合存放在内存中。如图1-4所示。

五、块

块是连续的多个字节或字的更长的组合。一般来说，块是没有固定长短的。但是在表示磁盘上的存贮大小时，块是有固定长度的，常用的块的长度是256字节、512字节或1024字节。块常用来存贮字符串，文本段或者程序段。

从一定意义上来说，数据类型反映了客观的需要，不同