8088/8086
汇编语言程序设计

李兆凤 编

8088/8086

中央广播电视大学出版社

8088/8086 汇编语言程序设计

李兆凤　编

中央广播电视大学出版社

（京）新登字163号

8088/8086汇编语言程序设计

李兆凤　编

中央广播电视大学出版社出版
新华书店北京发行所发行
北京市联华印刷厂印装

开本787×1092　1/16　印张22.5　千字513
1993年10月第1版　1994年10月第2次印刷
印数12001～28000
定价13.25元
ISBN 7-304-00835-0/TP·39

前　言

汇编语言是一种面向机器的语言。它能够利用计算机所有硬件特性并能直接控制硬件，在微型计算机系统的开发应用和过程控制中特别受到重视。

考虑到国内当前广泛使用8088/8086和80×86系统计算机的实际情况，本书以8088/8086系统的汇编语言作为学习的对象。学习了8088/8086系统的汇编语言程序设计后，掌握更高层次的汇编语言也就有了基础。

为了能为更多的读者服务，本书在编写中特别注重由浅入深、循序渐进并多举实例，使读者通过自学也能掌握程序设计方法。因此，本书适用于高等院校及成人教育的计算机应用专业，也可以作为其它专业学习计算机知识的教学参考书。

本书共分十章。第一章到第四章介绍了8088/8086系统结构、指令系统和汇编语言；第五章到第九章介绍了各种类型的程序设计，特别突出了非数值处理、输入/输出及中断程序设计；第十章介绍了系统调用及其程序设计。

汇编语言的程序设计能力，只有通过亲自实践才能提高。为此，与本书配套编写了《8088/8086汇编语言程序设计实验指导书》和《8088/8086汇编语言程序设计学习指导书》。其中，实验指导书讲述了上机操作的方法、步骤及实验题目，介绍了一些应用程序，提供了部分程序清单；学习指导书主要讲述了各章节中应掌握的内容及程度，提供了大量例题及习题的解题思路、方法和过程。这些内容将有助于对本教材的理解和程序设计能力的提高。

本书除第七章由刘晓星同志编写外，其它各章均由李兆凤同志编写。本书在编写过程中，得到北京邮电学院汪雍教授、北方交通大学刘颖荪教授、北京理工大学高永峰副教授的热情关怀，并得到他们提供的宝贵意见，在此表示衷心感谢。

编　者

1993年1月

目　录

第一章 概 论

计算机科学技术的飞速发展，已经使计算机应用在国民经济的各个方面。它对社会的生活、发展和进步起着巨大的影响。

作为计算机工作人员，除会使用高级语言编程，实现信息管理之外，为了开发出更有效的计算机应用系统或实时控制系统，学习和掌握汇编语言是非常必要的。

本章将就汇编语言及其程序设计的有关概念、特点进行讲述，以为后面各有关章节的学习作出必要的准备。

第一节 引 言

一、几个基本概念

1. 机器指令

机器指令是能为计算机所接受的一组二进制代码。它指出计算机所要进行的操作，以及其操作的对象（数据或数据地址）。它经过指令译码器译码，按照一定的时间周期使计算机动作起来以完成规定的操作。每一组代码构成一个机器指令。

为使计算机完成不同的操作，可以有不同的机器指令。把所有的机器指令集中在一起，就构成计算机的机器指令系统。它是计算机完成复杂操作功能的基本指令系统。

机器指令是由二进制代码组成的（因为计算机只能识别“1”、“0”二个状态）。一个机器指令有操作码部分和操作数部分。操作码指出该指令的功能，所要完成的操作。操作数指出操作的对象。它们分别用二进制代码表示。一般机器指令长度为单字节或多字节。一个字节为 8 个二进制位（8bit）。每个指令具体占几个字节，对每个指令均有规定。

2. 机器指令程序

要想使计算机完成一个有意义的操作，就应该用不同的机器指令编写出一系列机器指令的集合，即机器指令程序。交由计算机逐条执行，才能完成所要求的操作。

由于机器指令都是二进代码组，很难记忆，编出的机器指令程序也很难看懂，所以实际使用很困难，因而使用受限。现在只是在简单的计算机系统中或对计算机进行检测时，才使用一下机器指令程序。

3. 汇编指令

机器指令的上述缺点，迫使人们使用一种便于记忆的形式，即用助记符的方法来表示机器指令，称为汇编指令。这些助记符便于人们阅读和记忆，有点像大家所熟知的高级语言的味道，但又不一样。比如完成加法的指令可用 ADD 表示。而不能像高级语言直接进行二个数加法运算。

每一条机器指令对应一条汇编指令，所有汇编指令的集合就构成了计算机的指令系统。汇编指令又称符号指令，因为它是用助记符号来表示操作码和操作数的。

4. 汇编语言程序

用汇编指令编写的程序称为汇编语言程序。它是由汇编指令按照一定的语法规则编写的指令序列。严格说来，汇编语言程序不仅包括汇编指令，还应包括计算机完成规定任务所需要的数据和结果的存储地址以及组织程序所需要的指示性的伪指令（它们是非执行的指令）。

二、汇编语言

语言是交流的工具。同计算机进行信息交流，同样要使用“语言”。使用汇编指令系统编写程序就可以与计算机交流。所以称汇编指令系统及汇编伪指令，加上编程的语法规则就成为汇编语言。

从语言角度看，汇编指令就是汇编语言的指令语句。伪指令就是汇编语言的指示语句。

汇编语言比机器语言（机器指令系统）更容易理解、记忆和便于交流。然而它不能为计算机所直接接受。必须经过一次翻译，将其翻译为机器语言才能为计算机所识别。我们将完成这一翻译任务的程序叫做汇编程序。它是系统预先提供的系统软件之一，是专门用来完成把汇编语言程序翻译为机器语言程序这一任务的。

通常，把汇编程序的操作过程称为汇编。平常所说把汇编语言程序进行汇编，指的就是在汇编程序控制下，将汇编语言程序翻译为机器语言程序的过程。

由于汇编语言的指令语句与机器指令是一一对应的，所以它仍然是依赖于计算机的。因为计算机不同，计算机的机器指令系统就不同。本质上说，它仍然是一种面向机器的语言。它要求使用者对计算机本身要有一定的了解，并对计算机的指令系统要熟悉。这就给学习汇编语言带来一定的困难。

特别应该指出的是：汇编语言的指令语句与计算机程序设计人员所要完成的任务有很大的差别。由于指令语句所能实现的都是一些简单的操作，如累加器内容移位、数据在存储器与寄存器间的传递，寄存器内容的相加等，远非所要求的运算。要完成复杂操作，必须由程序设计人员把这些操作转为一系列汇编语言的指令语句。这种转换工作是有一定困难和耗费精力的，这就是为什么后来会发展出各种更乐意为人们所接受的高级语言的原因。

三、高级语言

高级语言是一种脱离具体计算机，面向过程，更符合人们思维和更易为人们所理解、学习的语言。如经常使用的 BASIC、FORTRAN、C、PASCAL、COBOL 语言等。它们的语句和逻辑结构与人们的思维基本上是一致的。所以用高级语言编程就变得容易而方便了。通常用高级语言编写程序要比汇编语言编写程序快 10 倍。而在调试程序方面，高级语言会更简单而节省时间。现在使用的大部分应用软件均是用高级语言编写的。它不依赖于计算机的类型。现在，已经有了各种国际标准语言版本供参照使用。用某种高级语言所编写的程序，一般在不同计算机上都可以运行（只要该机具有这种语言的编译程序就行）。

四、两种语言比较

概括地说，高级语言有下面一些优点：

1. 使用高级语言可以更容易、更快速地编写程序。编写出的程序也容易为它人阅读，即可读性好。

2. 使用高级语言编写的程序在调试中容易查错，使调试工作变得比较简单。

3. 一般高级语言都带有文件操作功能，可以将有关数据以文件方式存入外存储器中。

4. 高级语言不涉及具体的计算机体系结构及其指令系统。因此程序设计人员可以不必了解计算机的任何特性，而把精力集中在他们自己所要完成的任务本身工作上。

5. 高级语言编写的程序可移植性好。它们可以被移植到有该语言的编译程序的任何计算机上使用(因编译系统的不同，可能有部分语句需要修改)。

按照上面的说明，似乎高级语言最好，对使用者又容易又方便，为什么还要学习汇编语言呢？事物总是一分为二的。对计算机语言也不例外，有其长处必有其不利之处。

高级语言的缺点：用高级语言编写的程序，计算机不能直接执行，必须先把用高级语言编写的源程序，进行翻译，将其翻译为机器语言，我们把完成这一任务的翻译程序称为编译程序。由它将高级语言程序翻译成用机器指令表示的目标程序(二进代码模块)，目标程序才能为计算机所执行。高级语言设计时，是以面向过程而设计的，主要考虑怎样容易编写程序，以解决应用问题。因此编译时，就不可避免地出现所编译后的目标程序不够精练、过于冗长等问题。以致加大了目标程序的长度和增加了运行的时间。换句话说，将要占有较大的存储空间和较长的运行时间。无论从哪方面讲，都是不经济的。所以，高级语言并不能产生有效的机器语言程序。而汇编语言，则恰好能弥补高级语言的不足。它的指令语句与机器指令一一对应，由其编写的程序较为精练，一般来说会有较高的效率。所以在某些方面，如实时控制等，汇编语言仍然是受到人们青睐的一种语言。

可以说，在信息管理和科学计算方面使用高级语言较为适宜。在实时控制和专用微机系统中，使用汇编语言较为适宜。

第二节　汇编语言程序设计

一、汇编语言程序设计特点

汇编语言的特点已如上述，它居于机器语言和高级语言之间。不同的是，每个厂家的计算机均有自己的汇编语言。

汇编语言程序设计与高级语言程序设计有相似之处，但有很大的不同。下面通过一个具体程序例子，来看如何使用汇编语言编程。

如已知三个数，分别放在A、B、C三个单元中，求(A+B)*C的结果存放到D单元中。

对于用高级语言实现这一要求，是很容易的事，只要一个语句就可以完成了：

$$D=(A+B)*C$$

上述表达式的写法，对不同的高级语言可能有所不同，但其表达方式大体相似。这同我们通常的数学表达方式基本是一致的。显然，它直观、易懂、易记。但用汇编语言编写，则完全不同。它必须通过一些指令语句才能实现。

下面分别给出实现上述运算的流程图和源程序。流程图如图 1-1 所示。

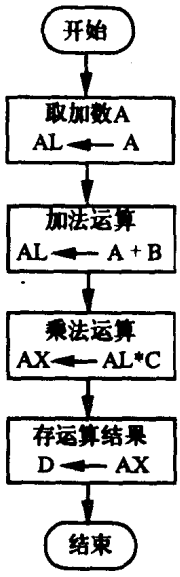

图 1-1　运算程序例流程图

```
DATA    SEGMENT
A       DB   05
B       DB   07
C       DB   0AH
D       DW   ?
DATA    ENDS
STACK   SEGMENT PARA STACK 'STACK'
        DB 100 DUP(?)
STACK   ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA,SS:STACK
ST:     MOV AX,DATA
        MOV DS,AX
        MOV AX,STACK
        MOV SS,AX
        XOR AL,AL
        MOV AL,A
        ADD AL,B
        MUL C
        MOV D,AX
CODE    ENDS
        END ST
```

从上述程序例中，可以看到完成同样功能，使用汇编语言程序设计要较高级语言程序设计“繁锁”得多。而且两种语言有着非常不同的思路。从某种意义上讲，汇编语言更讲究编程技巧。同一问题，由不同人员编写，可以写出完全不同的程序来，而其结果是相同的。

下面结合汇编语言程序设计的特点，谈谈学习中的注意事项，以供读者参考。

1. 汇编语言程序设计与计算机硬件有密切的关系。因此，为了掌握汇编语言程序设计，必须要较好地了解处理机的结构、性能及其有关部件的功能。如果要与外部设备进行数据传输，还要了解有关接口电路的功能。

2. 使用汇编语言编程时，对指令语句的选择是很重要的。同样的功能，使用不同的指令语句，可以有很大的差别。有经验的编程人员可以编写出占用存储空间少和执行时间短的程序，使得程序有较高的效率。所以在学习指令系统时，就应对每条指令的功能、作用有较深刻的理解并能熟记，以便在编程时能够正确地运用。

3. 在编写稍复杂一点的程序时，不要用指令语句直接编写程序。要先画出完成指定任务的

程序流程图，经检查无误后，再按流程图编写程序。这样编写出的程序逻辑错误较少。即使发现错误，根据流程图查错也很方便。所以，养成利用程序流程图编程的习惯是必要的。

4. 不管多么熟练的程序设计者，也不能保证编写出的程序绝对无错。因此，经过计算机执行程序以检验结果，是检查程序正确性的最好手段。学习计算机语言，要想学到程序设计的技能和技巧，只能多编程，多上机，也就是多实践。所以有人说：计算机学的好坏与上机时间成正比。

5. 程序设计是一件繁锁而又耗费脑力的事。特别是汇编语言程序设计更是如此。因此初学者一定不要怕学习中的困难，只要坚持多编程、多上机，就会逐步进入较为自由的境地，那时就会发现，程序设计是件非常有趣的事。

6. 通过汇编语言程序设计的学习，应该掌握一些经常应用的较成熟的子程序，以便今后编程需要时，加以利用。

7. 通过汇编语言程序设计和上机操作，应该学会对微型计算机系统的使用。特别是了解操作系统所使用的系统调用，掌握它们的功能、使用条件，可以为程序设计带来很多方便。

总之，我们希望读者在开始学习程序设计时，就能对上面提到的事项给予充分注意。如果能认真做好习题，并能多做一些实验，相信一定会为学习者打下一个汇编语言程序设计的初步基础。祝愿每位读者都能取得好的学习成绩。

二、程序设计步骤

计算机程序设计的含义是确定解题的方式，对问题的解决方法作出描述，把解决问题的方法或过程写成计算机能够接受的语句序例，即程序。程序设计是由程序员根据提出的任务去进行的。

程序设计只是把解决问题的方法转化为程序，但并不一定就能成为一个实用的程序。所以程序设计不但应该讲究方法，而且要掌握一些基本程序设计步骤。一个程序从设计到实用要经过一系列过程才能实现。

汇编语言程序设计有自己的特殊问题。即把解决问题方法的描述，需要编程人员通过思维转化为汇编语言的程序——一组求解问题的语句序列。这种劳动往往要比使用高级语言编程花费更多的时间和精力。

我们将程序设计的有关步骤用图1-2来表示。

理解设计任务，要求对求解的问题有明确的理解和认识。应给出明确的说明文件。它是设计的基础。只有把要解决的问题理解正确，才能编制出符合要求的程序。

数学模型是在说明文件的基础上，确定如何求解问题，最好是以数学表达式来描述问题的解决方法。

有了数学模型，应该确定计算方法，也就是解决问题的具体方法。一般，同一数学模型可以有多个算法，应选取执行速度快、占用内存空间少的算法。

根据算法，可以进行算法描述。通常要考虑程序中所采用的数据结构和程序结构。对较小的程序，一般使用程序流程图就可以了。对较大的程序，则应考虑模块化程序设计方法。

有了算法描述，就可以编写程序了。程序编好后，首先进行人工检查，有错则修改，直到认

为无错为止。源程序经汇编，可以给出语法错误。经运行，根据结果可以发现逻辑错误，即程序设计没有反映出原设计要求。它们均需要重新修改程序。要想获得一个正确的程序，需要反复检查和修改，最后才能得到一个可以应用的程序。

一个较好的程序，应该是在完成同一任务时，有较快的执行速度与占用较少的存储空间，这就是评价程序的时、空观点。当然，一个程序的结构及其可阅读性也是评价一个程序的重要条件。

顺便强调一点：程序流程图在程序设计中是一个很有用的工具。特别是初学者，应该在编程中习惯于先画流程图，然后再根据流程图编程序。这样做才会得到较好的效果。

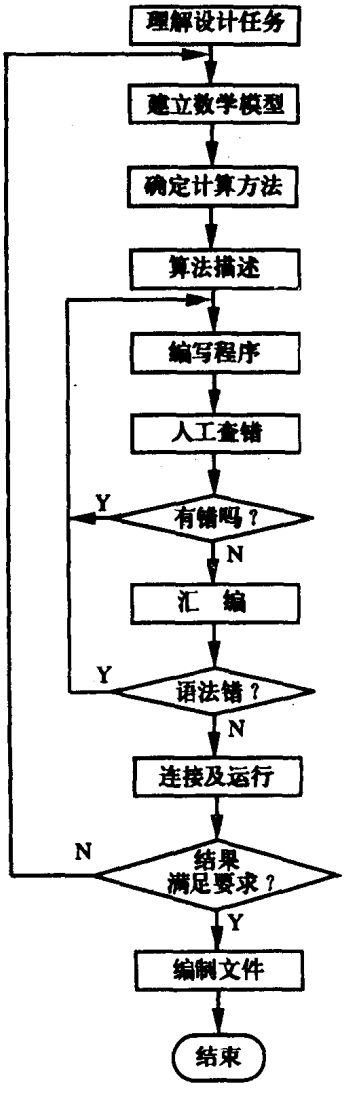

图 1-2 程序设计过程

第三节 汇编程序

一、汇编程序的作用

汇编程序是计算机系统软件之一。它的主要功能是将汇编语言程序转换为计算机能够识别并可执行的目标程序。

其工作过程可用图1-3表示。

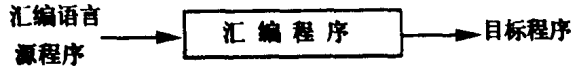

图 1-3 汇编程序功能图

通常，人们把用程序设计语言编写的程序称为该语言的源程序。它是由程序设计人员直接编写出来的。用汇编语言编写的程序，称为汇编语言源程序。把源程序送入到计算机中，计算机系统将调用汇编程序（对高级语言是编译程序）对源程序进行分析和处理。经过汇编程序处理，得到的是机器指令代码，即可以为计算机执行的目标程序。但其程序地址是浮动的。只有在加载运行时，才确定其在内存中的实际物理地址。

实际汇编后的结果。除目标代码之外，还有其它文件，如列表文件（源程序与目标代码清单）、交叉参考文件（符号索引表）等。它们可用来对程序进行分析和查找错误。

二、汇编过程

汇编程序为实现对汇编语言源程序的汇编，一般要借助于一些表格的帮助才能进行汇编。这些表格是设置在汇编程序中，在汇编时，通过对它们的检索和查询，才能进行汇编工作。

通常应有下面一些表格：

1. 机器指令操作代码表

供汇编程序查询要汇编的助记符指令的指令代码和指令长度。以便在汇编时，将汇编指令转为机器指令代码。

2. 伪指令操作表

汇编中，遇到伪指令时，汇编程序要完成伪指令定义功能的处理，如建立存储空间或数据存储区等。这种处理都是由伪操作程序段完成的。因此，该表给出伪指令及其处理程序地址。每遇到一个伪指令时，将转入该伪指令程序段中去执行，完成伪操作。

3. 符号表

符号表是在汇编过程中建立的。汇编语言允许使用符号(或标号)来表示地址和数据。它们出现在程序的标号段及操作数段，而且不一定定义在前，引用在后。因此在转换过程中，常遇到被引用的标号暂时无法确定其值。因此要先建立一个符号表。即将每个符号及其值或相对地址列入表中形成符号表。在汇编过程中，就可以查阅此表以获取每个符号的实际值代入要转换的机器指令代码中。

汇编程序为了实现对源程序各行的汇编，必须对源程序逐行扫描。把读到的语句行进行分析转换。

汇编程序一般都采用二次扫描完成汇编任务。第一遍扫描主要完成符号表的建立。它把所有的符号(变量、标号、常量等)及其值都集中在一起，形成一个符号表。

第二遍扫描才将每个程序行转换为机器指令代码或数据等。无论哪次扫描，都要使用一个地址计数器。每开始一个新的逻辑段，计数器清零，然后根据各指令码长或伪指令要求分配数据的单元数，对该计数器进行累加。计数器值就确定了符号的值(第一次扫描)或指令代码或数据的位置(第二次扫描)。

两次扫描的过程可用图1-4表示。

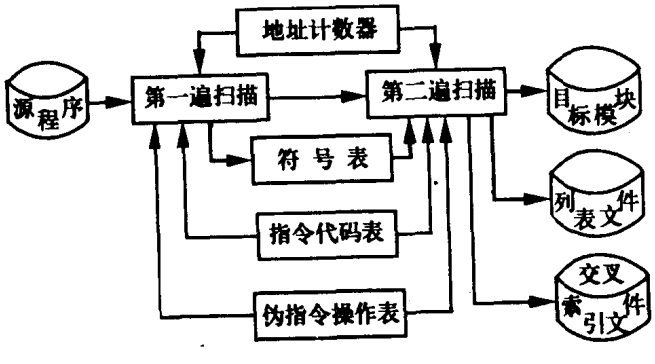

图 1-4　两次扫描汇编过程图

两次扫描都以遇 END 伪指令而结束。

在汇编过程中，还有一个重要的任务，就是检查源程序各语句行是否有语法错误。这是由汇编程序中的诊断程序实现的。当源程序的某个语句行不符合语法规则时，诊断程序将给出错误信息，以告诉程序设计人员进行修改。

以上就是汇编程序的简单工作过程。介绍的目的是使大家对汇编程序有一个基本的了解，以明了汇编程序的作用。