项目导向
任务驱动
培养技能
面向就业

全国高等职业教育计算机类规划教材·实例与实训教程系列

汇编语言程序设计

◎ 张绪辉　杜发启　主编
◎ 孙延维　陈　宇　副主编

- 以高级程序设计语言的视角全面诠释汇编程序设计语言
- 采用高级程序设计语言的架构建造汇编程序设计语言的框架
- 使用高级程序设计语言的例程演绎汇编程序设计语言的程序实现
- 辅以**模拟演示系统**透视用户汇编程序执行的全过程

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

全国高等职业教育计算机类规划教材·实例与实训教程系列

汇编语言程序设计

张绪辉　杜发启　主　编
孙延维　陈　宇　副主编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书以 Intel 8086/8088 CPU 为主，以 80X86 CPU 为扩展，以 DOS 操作系统为平台，以高级程序设计语言——C 语言为例，采用高级程序设计语言教程的模式，结合作者多年的教学经验，以大量实例，讲述汇编语言程序设计的方法和技巧。

汇编语言模拟系统可支持简单的汇编语言的程序设计，并可将用户设计的源程序翻译成机器代码，能模拟程序执行时的过程。该系统可帮助读者更好地理解汇编语言。

本书内容详细，通俗易懂，在章节安排上由简到繁，由浅到深。

本书配有电子教案和一个汇编语言模拟演示系统，可免费索取。

本书适合作为高职高专与相关专业的教材，也适合作为工程技术人员和自学者的参考书。

图书在版编目（CIP）数据

汇编语言程序设计 / 张绪辉，杜发启主编. —北京：电子工业出版社，2009.5
全国高等职业教育计算机类规划教材 • 实例与实训教程系列
ISBN 978-7-121-08643-4

Ⅰ. 汇… Ⅱ. ①张…②杜… Ⅲ. 汇编语言－程序设计－高等学校：技术学校－教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 058021 号

策划编辑：程超群
责任编辑：张燕虹
审　　校：于秀山
印　　刷：
装　　订：北京京师印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1 092　1/16　印张：17.25　字数：440 千字
印　　次：2009 年 5 月第 1 次印刷
印　　数：4 000 册　　定价：28.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

序

20世纪90年代以来，以计算机和通信技术为推动力的信息产业在我国获得前所未有的发展，全国各企事业单位对信息技术人才求贤若渴，高等教育计算机及相关专业毕业生供不应求。随后几年，我国各高等院校、众多培训机构相继开设计算机及相关专业，积极扩大招生规模，不久即出现了计算机及相关专业毕业生供大于求的局面。纵观近十年的就业市场变化，计算机专业毕业生经历了“一夜成名、求之不得”的宠幸，也遭遇了“千呼百应，尽失风流”的冷落。

这个时代深深地镌刻着信息的烙印，这个时代是信息技术人才尽情展示才能的舞台。目前我国的劳动力市场，求职人数过剩，但满足企业要求的专业人才又很稀缺。这种结构性的人才市场供求矛盾是我国高等教育亟待解决的问题，更是“以人为本，面向人人”为目标的职业教育不可推卸的责任。

电子工业出版社，作为我国出版职业教育教材最早的出版社之一，是计算机及相关专业高等职业教材重要的出版基地。多年来，我们一直在教材领域为战斗在职业教育第一线的广大职业院校教育工作者贡献着我们的力量，积累了丰富的职业教材出版经验。目前，计算机专业高等教育正处于发展中的关键时期，我们有义务、有能力协同全国各高等职业院校，共同探寻适合社会发展需要的人才培养模式，建设满足高等职业教育需求的教学资源——这是我们出版“全国高等职业教育计算机专业规划教材——实例与实训教程系列”的初衷。

关于本系列教材的出版，我们力求做到以下几点：

（1）面向社会人才市场需求，以培养学生技能为目标。工学结合、校企结合是职业教育发展的客观要求，面向就业是职业教育的根本落脚点。本系列教材内容体系的制定是广大高职教育专家、一线高职教师共同智慧的结晶。我们力求教材内容丰富而不臃肿、精简而不残缺，实用为主、够用为度。

（2）面向高职学校教师，以方便教学为宗旨。针对每个课程的教学特点和授课方法，我们为其配备相应的实训指导、习题解答、电子教案、教学素材、阅读资料、程序源代码、电子课件、网站支持等一系列教学资源，广大教师均可从华信教育资源网（www.huaxin.edu.cn）免费获得。

（3）面向高职学校学生，以易学、乐学为标准。以实例讲述理论、以项目驱动教学是本系列教材的显著特色。这符合现阶段我国高职学生的认知规律，能够提高他们的学习兴趣，增强他们的学习效果。

这是一个崭新的开始，但永远没有尽头。高等职业教育教材的建设离不开广大职业教育工作者的支持，尤其离不开众多高等职业院校教师的支持。我们诚挚欢迎致力于职业教育事业发展的有识之士、致力于高等职业教材建设的有才之士加入到我们的队伍中来，多批评，勤点拨，广结友，共繁荣，为我国高等职业教育的发展贡献我们最大的力量！

电子工业出版社高等职业教育分社

前　言

汇编语言在所有的程序设计语言中是一个“古老而又永葆青春”的程序设计语言。众所周知，计算机由两大系统组成：一是硬件系统，二是软件系统。计算机的运行是硬件系统根据机器指令而产生相应的硬件动作。用机器指令编程的效率太低，因此，计算机的前辈们就创造了汇编语言。汇编语言指令与机器指令有着一一对应的关系，至此，人们采用符号化的机器语言，即汇编语言进行编程，其编程效率大大提高。但是，由于汇编是面向机器的语言，其程序设计的方法与高级程序设计语言相比还是低效和难懂的，这决定了汇编语言难学。汇编语言是计算机能够提供给编程者的程序执行速度最快的语言，也是能够利用计算机硬件特性直接控制硬件设备的唯一语言，因而在对于软件的存储空间和运行效率要求很高的场合，汇编语言是最有效的编程工具。因此，汇编语言程序设计是高等院校计算机和相关专业的必修课。

汇编语言程序设计是微型计算机的接口技术、单片机等课程的先修课，一般学生在学习汇编语言程序设计之前至少已学习了一门高级语言程序设计，本书在介绍汇编语言时采用了高级语言的方法。

因为汇编语言与计算机结构紧密相关，所以本书在介绍汇编语言时以 8086/8088 CPU 为例介绍汇编语言的程序设计及相关知识。本书的使用者在学习本书内容前应具有扎实的 C 语言程序设计的基础。本书的实例都有 C 语言的实例、与 C 语言实例对应的汇编语言实例及优化了的汇编语言实例，这三种实例能够帮助学习者降低汇编语言程序设计的难度。

与本书配套的汇编语言模拟系统可提供汇编语言的编程设计，并将源程序翻译成机器指令，可视化模拟程序的执行过程。

本书共分 12 章，第 1 章、第 4～9 章由张绪辉编写，第 2 章由陈宇编写，第 3 章由孙延维编写，第 10～12 章、附录 A、附录 B、附录 C、附录 D 和习题由杜发启编写。

由于编者水平有限，书中难免会有欠妥之处，恳请广大读者提出宝贵意见。

编　者

2009 年 3 月

目　录

第1章 概 述

本章提要

- 三类语言的概念及其比较。
- 各种进制数的表示方法与相互转换方法。
- ASCII 码与 BCD 码。
- 数值数据与字符数据的计算机表示。

计算机程序设计语言是人与计算机交流的工具，到目前为止，计算机语言已经由低级到高级经历了机器语言、汇编语言、高级语言的发展过程。在纷繁众多的程序设计语言中，除机器语言外，汇编语言是唯一面向机器硬件结构与特性的语言，因此，对于学习和开发计算机软件的人们而言，通晓汇编语言程序设计是非常重要的。虽然今天的各种高级程序设计语言，如 C、VC++、Java，能够实现绝大多数机器语言可以实现的功能，但汇编语言还是经常被用来改进软件和硬件控制系统的效率；而对于动画游戏和虚拟现实应用软件来讲，汇编语言是唯一的编程语言选择，因为只有汇编语言才能提供高速度、高效率的代码。

汇编语言是一种面向机器的语言，汇编指令与机器指令具有一一对应的关系，对于 CPU 不同的计算机，其汇编语言也不同。因此，学习汇编语言必须首先了解对应于该汇编语言的计算机的硬件结构、数据类型及其在机器内的表示方法。

1.1 计算机语言是人机交流工具

计算机的工作是在人的操纵和控制下进行的。要完成这种操纵与控制，即让计算机按人的意愿工作，就必须与计算机交流信息，这个交流信息的工具就是计算机语言。目前使用的计算机语言分为三类：机器语言、汇编语言和高级语言。高级语言接近于自然语言，易学、易记，便于阅读，容易掌握，使用方便，通用性强，并且不依赖于具体的计算机。但是，计算机并不识别高级语言，它只能识别在设计该机器时事先规定好的机器指令。

1.1.1 机器语言

因为人们最初研制计算机的主要目的是进行科学计算，所以必须把数制引入计算机，也就是说要用电子器件来表示数制中的数码。人们习惯采用十进制，但是，要引入十进制，就必须制造具有 10 个稳定状态的电子器件，以表示十进制中的 0～9 这 10 个数码，这显然是不可能的，因为人类至今还没有研制出具有 10 个稳态且能够快速转换的器件。

人们研制、应用具有两个稳定状态且能够快速转换的电子器件技术却十分成熟，其应用也十分广泛，如开/关、通/断、亮/灭、高电平/低电平，用开、通、亮、高电平的状态表示二进制的数码 1，用关、断、灭、低电平状态表示二进制的数码 0，可很容易地实现状态的表示和状态转换。这样，二进制就自然而然地引入计算机中。这也是迄今为止计算机只认识 0、1

代码的根本原因所在。

由于计算机只认识0、1代码，所以人们与计算机进行信息交流时告诉计算机做什么、怎样做，就只能用一组0、1代码表示。指示计算机做什么、怎样做的一组0、1代码，称为机器指令。

计算机所具有的各种机器指令的集合，称为计算机的机器指令系统，又称为机器语言。用机器语言编写的程序称为机器语言程序。机器指令一般由操作码和操作数两部分组成：操作码部分告诉计算机做什么，即表征机器指令的基本操作功能，如加法、减法、传送等；操作数部分告诉计算机怎样做，即表征指令的操作对象或操作对象所处的地址形式。机器指令的一般形式如下：

操作码	操作数

Intel8086的机器指令是多字节指令，一条指令可以由1～7个字节组成。例如，如果要完成将偏移地址为100H的字节存储单元中的内容加上立即数10H，再将其和送回到地址为100H的字节存储单元的操作，则完成该操作的Intel8086机器指令由如下5个字节组成。

操作码	操作数	
	操作数1	操作数2
1000001100000110	0000000000000001	00010000

这5个字节在内存中如下存放：

10000011
00000110
00000000
00000001
00010000

其中，操作码占2个字节，即第一、二行中的两个8位二进制数是操作码，表示要进行“加”操作，还指明了以何种方式取得两个加数。第三、四行中的两个8位二进制数指出了第一个加数（称为目的操作数）所存放的偏移地址是100H，相加的结果也送入该存储单元中。第五个字节指出第二个加数（称为源操作数），是立即操作数10H。

机器指令也常常称为硬指令，它是面向机器的，即每台计算机都规定了自己所特有的机器指令。对于同样的二进制序列，不同型号的CPU对它的“理解”是不一样的。例如，上面的那一行指令代码在8088/8086 CPU看来是要求做加法，换到另一种CPU中完全可能被当成是另一种操作，甚至是错误的指令，所以机器代码与机器本身有着紧密的联系。每一种计算机（准确地说是每一种CPU）都有自己的一套指令，一种机型的所有机器指令的集合就是它的指令系统。

另外，从机器指令中可看出，任何一个操作数，均存放在一个具体的位置，操作这个数据必须知道该数据存放在什么位置和有多少个字节，而高级程序设计语言操作数据只需知道该数据的符号即可，这是机器语言与高级语言的区别之一。

1.1.2 汇编语言

由于机器指令是用0、1代码表示的，所以编写、调试和阅读都十分困难，极大地影响了计算机的应用和普及。于是，人们设想用符号来表示机器指令中相应的0、1代码的组合，即

用助记符来表示操作码，用变量、标号来表示内存存放的数据、指令的地址，用寄存器名来表示寄存器的编号，等等，就可以克服机器指令的缺点。

1.1.1 节的机器指令例子对应的汇编指令是：

```
ADD  WORD  PTR  [100H], 10H
```

在该指令中，助记符 ADD 表示加法操作；符号[100H]表示第一个加数（目的操作数）存放在数据段偏移地址为 100H 的地方；伪操作符 WORD PTR 则指明该目的操作数的类型为一个字（16 位二进制数），即目的操作数存放在数据段偏移地址为 100H 的连续两个字节中；立即数 10H 在指令中直接给出，即存放在该指令的第 5 个字节中，它为第二个加数（源操作数）；两个加数相加的和存放在数据段偏移地址为 100H 的内存单元中。

这种用符号书写的，其操作与机器指令基本上一一对应的，并遵循一定语法规则的计算机语言称为汇编语言。

用汇编语言编写的程序，称为汇编语言源程序。汇编语言也是面向机器的语言，对于不同的机器，其汇编语言也是不同的。学会 Intel 80X86 的汇编语言后，再学习其他机器的汇编语言也就容易多了。

由于汇编语言是为了方便用户编写程序而设计的一种符号语言，而计算机只认识 0、1 代码，因此，用汇编语言编写的程序并不能被计算机识别，必须将用汇编语言编写的程序翻译成由机器指令组成的程序后，计算机才能识别并执行。

要想让计算机识别，就必须先把汇编语言源程序翻译成机器语言程序（又称为目标程序），再通过连接程序（LINK）连接，生成可执行程序，才能被计算机识别并执行。

1.1.3 高级语言

虽然汇编语言用符号表示替代机器指令，从一定程度上简化了程序的编写、阅读和调试，但没有经过汇编语言程序设计训练的人员，编起程序来还是十分困难的，不利于计算机的进一步推广和应用。于是，人们尝试设计一种脱离具体的计算机、面向过程、更符合人的思维和更容易被人们所理解和学习的语言（称为高级语言）。20 世纪 50 年代末，第一个主要用于科学计算的高级语言——FORTRAN 语言诞生了。随后，各种高级语言如雨后春笋般地涌现出来。目前，世界上使用的高级语言有数百种，而且具有高功效的新的高级语言每天都在诞生。

由于计算机只能识别执行由 0、1 代码组成的机器语言程序，所以各种高级语言的源程序只有经过相应的编译程序或解释程序翻译成目标程序，经过连接程序连接后，才能被计算机识别并执行。完成这种功能的程序称为编译程序或翻译程序。编译程序和翻译程序的处理方式是不同的，但都完成将高级语言的源程序翻译成目标机器语言程序的功能。

1.1.4 三种语言的特点比较

从前面有关三种程序语言的介绍来看，编辑程序一般不采用机器语言，汇编语言代替了机器语言。一般编辑程序只使用汇编语言和高级语言，尽管高级语言有很多种，高级程序语言在人的思维模式、阅读和理解方面比汇编语言好，但仍不能代替汇编语言，汇编语言是一种青春常在的语言。

机器语言是由 0、1 代码组成的面向机器的语言。机器语言程序的编写、阅读和调试都十分困难，但它是计算机可直接识别并执行的语言程序，占内存少，执行速度快。

汇编语言是用符号表示替代机器指令的面向机器的语言。与机器语言相比，汇编语言易

于理解和记忆，汇编语言程序也易于编写、阅读和调试；由于其语句与机器指令语句一一对应，所以具有占内存少、执行速度快的特点，并且能直接控制计算机的硬件设备，充分发挥计算机的硬件功能。从本质上讲，汇编语言仍然是面向机器的语言，这就要求程序设计者要了解不同的计算机和熟悉不同的计算机的指令系统，这就给学习汇编语言带来了困难。另外，汇编语言的指令语句与程序设计者所要完成的任务有很大的差别。指令语句能实现的操作都是一些简单的操作，如寄存器数据的移位、寄存器与内存之间的数据传递、寄存器间的数据逻辑运算等，远非程序设计者要完成的任务。例如，当程序设计者要完成 *A=B+C+D* 的任务时，程序设计者必须将该任务转换为一条条操作动作的汇编指令：第一步将变量 *B* 在内存单元中的数据取到某个通用寄存器中；第二步将该通用寄存器的数据与变量 *C* 在内存单元中的数据取到算术逻辑运算部件进行加法运算，并将运算结果送到该通用寄存器；第三步将该通用寄存器的数据与变量 *D* 在内存单元中的数据取到算术逻辑运算部件进行加法运算，并将运算结果送到该通用寄存器；第四步将该通用寄存器的数据送到变量 *A* 所在内存单元中。

在执行汇编指令时，必须知道指令在什么地方执行，指令操作的操作数存放在何处，其汇编程序的指令序列不仅如人的思维逻辑过程，而且也如人的计算操作过程。例如对于一个简单的计算公式 2+3×4，人们可以很快地算出该式的结果是 14，但是对于更复杂的计算公式 567+123×56，人们就不可能很快地算出该式的结果，至少要借助笔和纸，首先计算 123×56=6888，并将结果 6888 写在纸上，然后再计算 567+6888=7455，并将结果 7455 写在纸上。简单的计算公式 2+3×4 对于一个只有小学文化水平的人而言，同样是一个复杂的计算公式，同样要用笔和纸，同样要按四则运算的规则计算。计算这样的公式，汇编语言必须反映这样的逻辑思维过程和操作过程，只不过计算部件是算术逻辑部件（ALU），其执行的每一步的结果必须存放在通用寄存器或内存中，其执行的汇编指令序列如下：

```
MOV     AL, 123         ; 将操作数 123 送通用寄存器 AL
MOV     BL, 56          ; 将操作数 56 送通用寄存器 BL
MUL     BL              ; 将通用寄存器 AL 和 BL 的值相乘，其积送通用寄存器 AX
ADD     AX, 567         ; 将通用寄存器 AX 的值加 567，其和送通用寄存器 AX
```

这种运算方式是非常细致和烦琐的，这也是人们研发高级语言的原因。

例如 ABC=567+123×56，在高级语言中的语句就是如此，它反映了人们的逻辑思维过程，至于 567+123×56 如何具体计算，其结果如何送到 ABC 中，ABC 在何处，均不需要考虑。高级语言的主要特点是：脱离具体的机器，面向过程，是一种类似于自然语言和数学描述语言的程序设计语言。高级语言程序易于编写、阅读和调试，并且可移植性好。

因为高级语言源程序经过编译后才能翻译成计算机可识别、执行的目标程序，所以不能产生有效的机器语言程序，其运行速度较慢，占内存较大。汇编语言程序正好弥补了这一缺点。

各种语言的特点如表 1.1 所示。

表 1.1　各种语言的特点

机器语言的特点	汇编语言的特点	高级语言的特点
与机器有关	与机器有关	与机器无关
面向操作	面向操作	面向过程
编程难	编程一般	易于编程
阅读、理解、记忆难	阅读、理解、记忆一般	易于阅读、理解、记忆

续表

机器语言的特点	汇编语言的特点	高级语言的特点
内存可控，占用少	内存可控，占用少	内存不可控，占用多
运行效率高，速度可控	运行效率高，速度可控	运行效率低，速度不可控
应用于控制系统、实时系统、系统软件	应用于控制系统、实时系统、系统软件	应用于非实时的应用系统

1.2 汇编语言的架构

1.2.1 汇编语言源程序的架构

汇编语言是较早发明的一种介于自然语言与机器语言之间的程序设计语言。为了使汇编语言到机器语言的翻译比较简单，汇编语言用大量的语法规则对从指令到程序的书写加以限制。与后来的高级语言相比，汇编语言更接近于机器语言，用汇编语言编写的源程序还保留了很多机器语言的特点。例如：

机器指令中的操作码部分在汇编语言中用与该指令的功能相关的一个符号表示，如加法指令就用 ADD 表示，数据传送指令用 MOV 表示，这类符号称为“助记符”。

对于一个放在内存中的操作数，如果要求编程人员记住每一个操作数在内存中的存放位置，则是一个巨大的负担。在汇编语言中，减轻这种负担的方式是用变量存放操作数，程序员只需记住变量的名字。

跳转是程序设计中不可避免的一个问题。在机器语言中，跳转的目的地是用指令所在位置与当前跳转指令的相对位置来表示的，而汇编语言中的跳转则是在目的地做一个称为“标号”的标记。

除了与机器语言有直接对应关系的助记符、变量、标号外，为了能让汇编程序正确地完成翻译工作，必须告诉汇编程序变量需要占据多少字节的内存、程序到何处结束、整个程序的第一条指令在什么地方等问题。因此，源程序中应该有一些告诉汇编程序如何进行翻译操作的“说明”，这类说明在翻译结果中并没有对应的机器指令，所以称为“伪指令”。

下面通过一个例子来了解和分析汇编语言源程序格式。

【例 1.1】 试编写一个程序，将变量名为 BOY 的内存字单元中的二进制数转化成十六进制数，并在屏幕上显示出来。

```
DATA    SEGMENT                 ; 定义一个段名为 DATA 的数据段，并在 BOY 字单元中提供
                                ; 一个待转化的 16 位二进制数
        BOY   DW   0101100010101111B
DATA    ENDS
CODE    SEGMENT                 ; 定义代码段
        ASSUME DS:DATA,CS:CODE
START:  MOV   AX,  DATA
        MOV   DS,  AX           ; 数据段寄存器赋初值
        MOV   AX,  BOY          ; 取二进制数送累加器 AX
        MOV   CH,  4            ; 循环次数置初值
        MOV   CL,  4            ; 设移位次数
LOOPl:  ROL   AX,  CL           ; 转化成十六进制数
```

```
        MOV   BX,   AX
        AND   AL,   0FH
        CMP   AL,   09          ; 转化成 ASCII 码
        JBE   DCM
        ADD   AL,   7
DCM:    ADD   AL,   30H
        MOV   DL,   AL          ; 显示该位十六进制数
        MOV   AH,   2
        INT   21H
        MOV   AX,   BX          ; 恢复中间数据
        DEC   CH                ; 控制循环次数
        JNZ   LOOP1
        MOV   AH,   4CH
        INT   21H               ; 返回 DOS
CODE    ENDS                    ; 代码段结束
        END   START             ; 该源程序结束并指明执行的第一条指令
```

该程序由堆栈段、数据段、代码段组成。

1.2.2 汇编程序和连接程序

汇编源程序必须经过汇编程序翻译成机器语言程序，然后经过连接程序装配成可执行程序。汇编程序和连接程序属于软件分类中的系统软件部分。不同的机器具有不同的机器指令，也就具有不同的汇编语言，同样具有不同的汇编程序和连接程序。

汇编程序MASM.EXE是一种专门用于把Intel 8086/8088的汇编语言源程序翻译成相应的机器语言程序的翻译器，是 8086/8088 汇编语言编程人员必备的基本工具之一。汇编程序还具有语法检查的功能，交给汇编程序进行处理的源程序在翻译之前都必须经过语法检查这一关。如果汇编程序发现源程序中有不符合语法要求的指令，将不进行翻译工作，而是指出错误的位置及错误类型。不同厂家、不同版本的汇编程序在语法上可能有细微的差别。本书后面章节中都以 Microsoft 公司的 MASM.EXE V5.0 作为汇编程序，如果读者所使用的汇编程序不是这个版本，请参照有关说明书。

汇编程序翻译的结果已具备机器语言的形式，称为“目标程序”，一般以 OBJ 作为文件扩展名。但是，目标程序还不能直接交给计算机去执行，它还需要通过连接程序（LINK.EXE）的装配才具备可执行的形式，装配结果称为“执行文件”，一般以 EXE 作为文件扩展名。另外，连接程序还具有把多个目标程序装配在一起的功能，或者把目标程序与预先编写好的一些放在子程序库中的子程序连接在一起，构成较大的执行文件。汇编语言源程序到目标程序的汇编过程如图 1.1 所示，目标程序到执行程序的连接过程如图 1.2 所示。

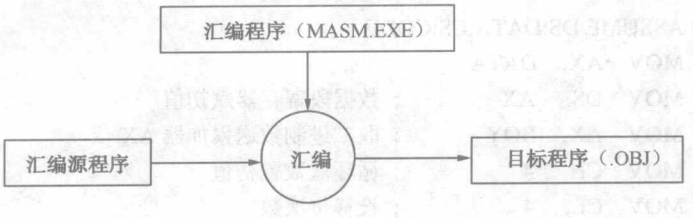

图 1.1　汇编语言源程序到目标程序的汇编过程