

免费提供
电子教案

高等院校规划教材
计算机科学与技术系列

汇编语言程序设计

何超 主编



机械工业出版社
CHINA MACHINE PRESS



高等院校规划教材·计算机科学与技术系列

汇编语言程序设计

何 超 主编
何 翔 胡安民 田桂丰 王煜林 方琳 参编



机械工业出版社

本书共 8 章, 主要讨论汇编语言的编程基础、程序开发过程和调试方法。
本书的主要特点是通俗易懂, 遵循由浅入深、由简到繁、循序渐进的原则。力争改变汇编语言难学难教的状况。

与本书配套的《汇编语言程序设计实验与习题解答》一书另行出版, 该书包含详尽的汇编语言程序设计实验和本书所有习题的解答。

本书既可作为高等院校信息类(如计算机、自动控制、电工电子等)专业的本科教材, 也可作为工程技术人员的参考书。

图书在版编目(CIP)数据

汇编语言程序设计 / 何超主编. —北京: 机械工业出版社, 2009.9

(高等院校规划教材·计算机科学与技术系列)

ISBN 978-7-111-27260-1

I. 汇… II. 何… III. 汇编语言—程序设计—高等学校—教材
IV. TP313

中国版本图书馆 CIP 数据核字(2009)第 082132 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 陈 皓

责任印制: 李 妍

北京汇林印务有限公司印刷

2009 年 10 月·第 1 版第 1 次印刷

184mm×260mm·22 印张·544 千字

0001—3000 册

标准书号: ISBN 978-7-111-27260-1

定价: 38.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

门户网: <http://www.cmpbook.com>

销售一部: (010) 68326294

教材网: <http://www.cmpedu.com>

销售二部: (010) 88379649

读者服务部: (010) 68993821

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

目前，汇编语言的书籍已多如牛毛，为什么还要冒很大的市场风险来编写这本书呢？

第一，汇编语言还很实用。计算机发展到今天，单纯的计算机应用知识已经普及了，而人们更加关心的是计算机在控制方面的运用。

汇编语言与计算机硬件联系密切，所以凡涉及与计算机硬件相关的输入输出和控制方面的程序，以及在单片机控制，嵌入式应用，仪器、仪表及家用电器中用于控制方面的程序等，大多是用汇编语言编写的。其中，单片机控制、嵌入式应用，已经或正在改变着社会生产和日常生活的方方面面。

学习汇编语言可以使读者了解计算机原理最基础的部分，为进一步学习操作系统、接口技术、单片机和嵌入式等后续课程打下坚实的基础。

第二，比较多的汇编语言书籍给人的感觉是很难懂。分析其原因，主要包括以下几点：

1) 汇编语言根植于计算机的硬件底层，涉及太多的机器指令和复杂规则，难于理解和综合运用。

2) 汇编语言体系错综复杂，盘根错节，难于理清。例如，一些汇编语言的书籍不太讲究知识的逻辑线索；在讲某知识时，插入了一些生僻的术语；行文未遵循由浅入深、由简到繁、循序渐进的认识规律。

3) 在对程序的解释上，一些汇编语言的书籍存在过于简单，或过于迂回，甚至张冠李戴的问题，从而让读者大费精力去猜测。

4) 在用语上，一些汇编语言的书籍偏于深奥和过于“专业化”，给读者增加了理解上的困难。

5) 手册式的知识罗列，枯燥的叙述模式，使读者不易入门，也很难感兴趣，以至于缺乏学习动力。

第三，上述情况已经引起了广泛关注。一时间，“汇编语言”课程难学、难教几乎成了“共识”。但也有不少新书，如王爽先生的《汇编语言》，温玉洁、梅广宇、罗云彬 3 位先生翻译的美国人 Kip.R. Irvine 著的《Intel 汇编语言程序设计》（第 5 版），梁发寅、宗大华两位先生的《汇编语言程序设计》等，都在力图改变这种状况。尽管这些书深度不等，适用层次不同，风格各异，但都给人以耳目一新的感觉。

基于以上几点，本书也将尝试加入到这一力争改变汇编语言难学、难教现状的行列中来，并在以下几方面作出努力：

1) 尽力理清汇编语言知识体系，在体系架构的设计上力求由浅入深、由简到繁、循序渐进。

2) 为了保证知识的完整和相互联系性，采用“知识屏蔽”的方法；实在绕不开的地方，采用此处加标题，其他处（直接相关或有用的地方）再详解的办法。

3) 表面上看，本书的体例和其他书大同小异，实则有较强的差别。例如，考虑到汇编语

言知识体系比较庞杂，本书第 1 章带有“巡礼”的意味，先带领读者“走马观花”地参观一下汇编语言“世界”，在以后各章中再加详述。第 1 章中，在“寻址方式”的编写上下了较大的功夫，力求深入浅出，化解这个学习难题。

第 2 章是第 1 章内容的自然延续，讨论了指令与伪指令，讲述了汇编语言的基本“游戏规则”。而且，伪指令在前，指令在后，这主要是考虑了知识的逻辑线索的需要，如指令是对数据操作的，而数据的定义如果尚未明白，指令是学不好的。

这样，前两章就为后面的学习打下了坚实的基础。

趁热打铁，第 3 章就讨论编程的基本知识，这是第 1 轮知识的循环；第 4 章深入到子程序的学习；第 5 章是专设的一章，讨论与理工类专业的有关问题的程序设计，这是第 2 轮知识的循环，这样安排，有助于读者及时消化所学知识，并达到学习效果的螺旋式上升。之后各章的学习就是顺理成章的事情了。

4) 考虑到语言学习的特点，应该从模仿起步。所以本书例题多，类型覆盖面广，解释详尽（当然，前后类似的地方也注意了减少赘述），尽量减少读者学习的拦路虎。

5) 本书的立足点是初学者的角度，注意用语通俗、生动活泼，但又不失科学性，术语一定先有解释，后再使用。术语的解释中绝不再出现另外生僻的词语。

6) 图文并茂，变看不见的事物为看得见的图示。

7) 为了降低学习的难度，并考虑到初学者的困难，本书给出的程序大多是完整的，并经过调试的。

下面具体介绍本书各章的内容。

第 1 章 汇编语言的基本概念。首先讲解了机器语言与汇编语言，然后不可避免地讨论了计算机中的编码和字符表示。接着讨论计算机 CPU 的编程结构，内存及 I/O 端口；又简介了汇编语言的源程序格式和语句类型；最后着重而详尽地讨论了计算机的寻址方式。

第 2 章 汇编语言程序中的指令与伪指令。本章是第 1 章内容的自然延续，讲述汇编语言的基本“游戏规则”。

第 3 章 汇编语言程序设计初步。在前两章的基础上开始讲解编程，讲解了 3 种基本的程序结构：顺序结构、分支结构和循环结构。

第 4 章 子程序设计。集中讨论了汇编语言中很重要的子程序及其嵌套。

第 5 章 汇编语言程序的应用示例。考虑到运用汇编语言的读者大多数与理工类的专业有关，所以专设本章讨论相关问题的程序设计。例如，数学运算、字符串与数组的处理、数据的分类与统计、代码转换及列表等问题的讨论。

第 6 章 输入输出和中断处理。本章讨论输入输出和中断处理方面的相关内容。

第 7 章 宏和多模块程序设计。本章既是全书内容的总结，又是全书内容的提高。其中，涉及高级汇编语言的设计问题，也涉及了汇编语言与高级语言接口的问题。

第 8 章 汇编语言程序的开发与调试。

鉴于面向 8086/8088CPU 体系的汇编语言相对简单，并且掌握了它，再学习其他体系的汇编语言也就不难了。所以本书以 8086/8088CPU 体系为背景编写。

本书由何超主编，参与编写的还有何翔、胡安民、田桂丰、王煜林和方琳。其中第 1、2 章由何超编写，第 3、5 章由何翔编写，第 4、6 章由胡安民编写，第 7 章由田桂丰和方琳编写，第 8 章由方琳编写，习题部分由王煜林编写。

裴方华为本书调试检测了大部分程序，在此表示感谢。

本书既可作为高等院校信息类（如计算机、自动控制、电工电子等）专业的本科教材，也可作为工程技术人员的参考书。

鉴于作者水平有限，书中不妥之处在所难免，敬请广大读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 汇编语言的基本概念	1
1.1 汇编语言概述	1
1.1.1 汇编语言与机器语言	1
1.1.2 汇编环境介绍	3
1.1.3 汇编语言上机过程	3
1.2 微型计算机中数的编码和字符的表示	4
1.2.1 进位计数制	4
1.2.2 进制之间的转换	5
1.2.3 无符号数和带符号数	8
1.2.4 定点数与浮点数	9
1.2.5 计算机中的编码	11
1.3 8086CPU、内存及 I/O 端口	13
1.3.1 内存	13
1.3.2 I/O 端口地址	18
1.3.3 8086 CPU 的编程结构	19
1.3.4 8086CPU 的寄存器	20
1.3.5 80x86CPU 的寄存器	23
1.4 汇编语言的源程序格式和语句类型	23
1.4.1 汇编语言的源程序格式和汇编语言的语句类型	23
1.4.2 汇编语言的源程序格式示例	25
1.4.3 符号地址	27
1.4.4 汇编语言可执行程序格式	28
1.5 汇编语言中操作数的寻址方式	29
1.5.1 操作数的寻址方式	29
1.5.2 操作数的寻址方式在指令中的运用	30
1.5.3 关于寻址方式的 3 点说明	35
1.6 本章小结与学习目标	36
1.7 习题一	36
第 2 章 汇编语言程序中的指令与伪指令	40
2.1 汇编语言的语句格式、运算符及表达式	40
2.1.1 汇编语言语句格式	40
2.1.2 操作数及其表达式	42
2.1.3 操作数表达式中的运算符	44

2.2	汇编语言程序中的伪指令	48
2.2.1	段定义伪指令	48
2.2.2	ASSUME 伪指令	48
2.2.3	数据定义伪指令	50
2.2.4	符号定义伪指令	51
2.2.5	用伪指令 LABEL 定义变量和标号	53
2.2.6	PTR 操作符、LABEL 伪指令与 THIS 操作符的区别	54
2.2.7	置汇编地址计数器伪指令	55
2.2.8	地址定位伪操作	56
2.2.9	基数控制伪指令	57
2.2.10	过程定义伪指令 PROC 和 ENDP	58
2.3	8086/8088 处理器的汇编语言常用指令	58
2.3.1	数据传送指令	58
2.3.2	算术运算指令	67
2.3.3	逻辑运算指令	81
2.3.4	移位指令	83
2.3.5	循环移位指令	86
2.3.6	串操作指令	88
2.3.7	与转移地址有关的寻址方式——无条件跳转指令 JMP	89
2.3.8	处理机控制指令	93
2.4	32 位新增指令简介	94
2.4.1	寻址方式的变化	94
2.4.2	32 位新增指令	94
2.5	本章小结与学习目标	99
2.6	习题二	99
第 3 章	汇编语言程序设计初步	104
3.1	系统功能调用	104
3.1.1	系统功能调用概述	104
3.1.2	输入输出 (I/O) 系统功能调用	105
3.2	汇编语言源程序的基本框架和设计的基本步骤	108
3.2.1	源程序的基本框架	108
3.2.2	源程序设计的基本步骤	111
3.3	顺序结构的汇编语言源程序的编写	111
3.4	分支结构的汇编语言源程序的编写	115
3.4.1	条件转移指令	115
3.4.2	汇编语言分支结构程序的编写	117
3.5	循环结构的汇编语言源程序的编写	124
3.5.1	循环控制指令	124
3.5.2	程序的循环结构	125

3.5.3	控制程序循环的方法	125
3.5.4	多重循环	134
3.6	本章小结与学习目标	136
3.7	习题三	137
第4章	子程序设计	140
4.1	子程序的定义和格式及设计步骤	140
4.1.1	子程序的定义和格式	140
4.1.2	子程序的设计步骤和子程序的说明文件	141
4.2	子程序的调用和返回	142
4.2.1	调用与返回指令	143
4.2.2	现场的保护与恢复	146
4.3	主程序与子程序之间传递参数的方式	147
4.3.1	寄存器法	148
4.3.2	约定存储单元法	151
4.3.3	堆栈传递参数法	155
4.4	子程序设计举例	159
4.5	子程序嵌套	162
4.5.1	子程序的嵌套	162
4.5.2	子程序的递归和可再入性	162
4.6	本章小结与学习目标	166
4.7	习题四	166
第5章	汇编语言程序的应用示例	170
5.1	数学运算的汇编语言程序示例	170
5.1.1	四则运算类程序示例	170
5.1.2	求一个数的平方根程序示例	172
5.1.3	求一个数的绝对值程序示例	174
5.1.4	求一个数的相反数程序示例	175
5.1.5	累加求和程序示例	177
5.1.6	计算一个复杂表达式的累加和——提高循环程序的效率示例	180
5.1.7	矩阵乘法程序示例	182
5.2	汇编语言的字符串	185
5.2.1	字符串及其操作指令	185
5.2.2	字符串的运算概述	188
5.3	数据的分类和统计程序	202
5.4	代码转换	215
5.4.1	进制转换	215
5.4.2	其他代码转换	218
5.5	列表	220
5.5.1	常数表	221

5.5.2	利用表实现码之间的转换	222
5.6	本章小结与学习目标	223
5.7	习题五	223
第6章	输入输出和中断处理	225
6.1	输入输出和中断的基本概念	225
6.1.1	I/O 方式概述	225
6.1.2	中断与中断分类	227
6.1.3	中断向量与中断向量表	229
6.1.4	中断指令	230
6.1.5	中断处理过程	231
6.1.6	中断优先级与中断嵌套	232
6.1.7	存储器直接访问 (DMA) 方式	233
6.1.8	I/O 处理机方式	234
6.2	BIOS 和 DOS 及其中断调用	234
6.2.1	DOS 中断	235
6.2.2	系统功能 INT 21H 的调用方法	236
6.3	常用 BIOS 中断调用	243
6.3.1	BIOS 中断的分类	243
6.3.2	常用 BIOS 调用	244
6.3.3	INT 16H (键盘 I/O 调用)	244
6.3.4	显示器调用	246
6.4	文件管理	259
6.4.1	DOS 磁盘管理	259
6.4.2	BIOS 磁盘存取功能	262
6.5	中断处理程序设计	264
6.6	本章小结与学习目标	269
6.7	习题六	270
第7章	宏和多模块程序设计	272
7.1	宏	272
7.1.1	宏定义	272
7.1.2	宏调用	274
7.1.3	宏展开	274
7.1.4	宏的参数传递	275
7.1.5	特殊宏操作符	276
7.1.6	宏体中的变量和标号	279
7.2	宏嵌套	281
7.2.1	宏定义中嵌套宏定义	282
7.2.2	宏定义中嵌套宏调用	282
7.3	重复汇编伪指令和条件汇编伪指令	283

7.3.1	重复汇编伪指令 REPT	284
7.3.2	重复汇编伪指令 IRP	285
7.3.3	重复汇编伪指令 IRPC	285
7.3.4	条件汇编伪指令 IF	286
7.4	宏汇编语言源程序举例和宏与子程序的区别	287
7.4.1	宏汇编程序举例	287
7.4.2	宏与子程序的区别	289
7.5	多模块程序设计	290
7.6	汇编语言与高级语言的连接	301
7.7	本章小结与学习目标	306
7.8	习题七	306
第 8 章	汇编语言程序开发与调试	308
8.1	汇编语言程序开发过程	308
8.2	汇编语言程序的调试方法	312
附录	318
附录 A	ASCII 码表	318
附录 B	Pentium 指令系统一览表	318
附录 C	BIOS 调用	323
附录 D	DOS 功能调用说明	332
附录 E	IBM PC 的键盘输入码和 CRT 显示码	338
参考文献	340

第 1 章 汇编语言的基本概念

本章提要

本章是全书的基础，也是理解任何计算机工作原理的基础，更是学习汇编语言的基础。首先介绍了机器语言和汇编语言的基本概念及学习汇编语言的重要性和必要性。接着说明了计算机中数的编码和字符的表示方法，这是计算机得以运行的基础；讲解了微处理器的编程结构，寄存器、存储器的逻辑结构与计算机中数据存取的寻址方式的密切关系。

本章重点讨论了数据的各种寻址方式，逻辑地址与物理地址的概念、求取和相互关系，以及堆栈和 I/O 端口地址的知识。

1.1 汇编语言概述

1.1.1 汇编语言与机器语言

1. 机器语言

学过计算机基础课程后，可以知道，指令是使计算机执行各种操作的命令。按照具体的运算功能要求，把一条一条指令排列和组织起来，构成程序。编写程序的方法和规则很多，形成不同的计算机语言。高级语言的语句形式接近于自然语言，比较易于学习和掌握。但不能被计算机直接识别和执行，必须先通过编译和处理，增加了计算机的工作量。

计算机通过 CPU 能直接识别和执行的指令，称为机器指令。机器指令与计算机的 CPU 和其他硬部件的具体结构有密切的关系，通常不同种类的计算机对应的机器指令也不同。在表现形式上，机器指令为二进制代码，机器语言是用二进制编码的机器指令的集合（简称指令集）及一组使用机器指令的规则。

机器语言是计算机能直接识别的唯一语言，但机器语言不能用人们熟悉的形式来描述计算机需要执行的任务，且编写程序十分麻烦，总是长长的一串 0 和 1，容易出错，调试也困难。

例如，用机器指令编写的两数相加的程序段，用二进制表示如下：

```
10100000 00000000 00100000
00100000 00000110 00000001 00100000
10100010 00000010 00100000
```

几乎没有人能直接看出上述程序的功能。可见，即使是程序员也难以掌握机器语言。

2. 汇编语言

汇编语言是一种面向计算机 CPU 指令系统的程序设计语言，它采用一组字母、数字或符号来代替一条二进制编码表示的指令，因而比机器语言易于读写、调试和修改。实际上，汇编语言就是机器语言程序的符号表示。

例如，在汇编语言中，用 MOV 表示数据传送；ADD 表示加法运算。

前述两数相加的程序段可以用下面的汇编语言程序段来表示（暂时看不懂程序不要紧，以后会逐步讲到）。

```
MOV  AX, DATA1    ; 将存储地址为 DATA1 的数， 移送到通用寄存器 AX 中
ADD  AX, DATA2    ; 将存储地址为 DATA2 的数， 与 AX 中的数（现在是 DATA1）相
                          ; 加， 运算结果放在 AX 中
MOV  DATA3, AX    ; 将寄存器 AX 中的数（即刚才的运算结果）， 移送到存储地址为
                          ; DATA3 的存储单元中
```

（1）汇编语言的优点

- 1) 面向机器的低级语言，直接控制计算机硬件部件。可有效地访问、控制计算机的各种硬件设备。
- 2) 保持了机器语言的优点，具有直接和简洁的特点。
- 3) 目标代码简短，占用内存少，执行速度快，是高效的程序设计语言。
- 4) 经常与高级语言配合使用，应用十分广泛。

（2）汇编语言的缺点

1) 与处理器密切相关，通常是特定的计算机或系列计算机专门设计的。汇编语言依赖于具体的处理器体系结构，不能通用，因此不能直接在不同处理器体系结构的计算机之间移植。不同种类的计算机对应的汇编语言不同，但大同小异。

2) 编写程序，要熟悉计算机硬件系统，要考虑与硬件相关的许多细节。在编写复杂程序时，汇编语言相对高级语言代码量较大，编写繁琐。调试、维护，也比较困难。

（3）汇编语言的应用场合

- 1) 程序要求具有较快的执行速度，或者只能占用较少的存储空间。
- 2) 程序与计算机硬件密切相关，程序要直接、有效地控制硬件。
- 3) 大型软件中需要提高性能、优化处理的部分。
- 4) 没有合适的高级语言，或只能采用汇编语言时。
- 5) 分析具体系统，尤其是该系统的低层软件、加/解密软件时，分析和防治计算机病毒时。

本书介绍的是对应于 80x86 CPU 的汇编语言，掌握了它，再学习对应于其他 CPU 的汇编语言，也就不难了。

3. 汇编程序

由于 CPU 能直接识别的语言是机器语言，用机器语言描述的程序称为目的程序（或目标程序），所以用汇编语言编写的源程序必须翻译成为用机器语言表示的目标程序后，才能由 CPU 执行。把汇编语言源程序翻译成目标程序的过程称为汇编，完成汇编任务的程序称为汇编程序，汇编过程如图 1-1 所示。

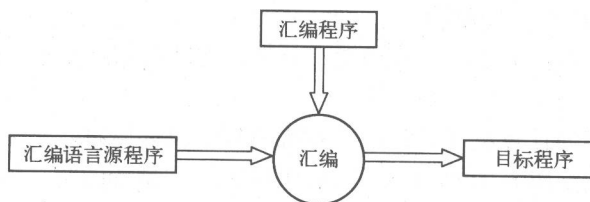


图 1-1 汇编过程示意图

常用的汇编程序（汇编编译器）有 Microsoft 公司的 MASM 系列和 Borland 公司的 TASM 系列。汇编程序以汇编语言源程序文件作为输入，并由它产生 3 种输出文件，即目标程序文件、源程序列表文件和交叉引用文件。其中，目标程序文件经连接定位后由计算机执行；源程序列表文件将列出源程序和对应的目标程序的机器语言代码清单及符号表，使程序调试起来更加方便；交叉引用文件用于产生交叉引用表。源程序列表文件和交叉引用文件是在这个过程中产生的中间文件，通常不需要建立这两个文件，对于计算机的屏幕提示，按〈Enter〉键即可。在本书中，不讨论它们。

❗注意：汇编语言源程序、目标程序和汇编程序三者的区分。

4. 为什么要学习汇编语言

由于汇编语言就是机器语言程序的符号表示，而机器语言是计算机能直接识别和执行的语言。因此，用汇编语言编写的程序比高级语言编写的程序简洁，占用内存空间少，运行速度较快。有资料表明，70%以上的系统软件是用汇编语言编写的，某些高级绘图程序、视频游戏程序，也是用汇编语言编写的。

计算机发展到今天，单纯的计算机应用知识已经普及了。人们更加关心的是计算机在控制方面的运用。汇编语言与计算机硬件联系密切，所以与计算机硬件相关的输入输出和控制方面的程序，以及在单片机、仪器、仪表及家用电器中用于控制方面的程序等，大多是用汇编语言编写的。学习汇编语言，必然要对计算机工作过程有更多的了解。这对学习微机原理及其接口技术、操作系统、嵌入式系统及微机控制技术等后续课程，都是有利的。况且，在这些课程中，也会接触到不少汇编语言编写的程序。所以学好汇编语言是很有必要的。

5. 学习汇编语言的知识准备

学习汇编语言之前，要掌握计算机基础类课程中所学过的基础知识，例如，计算机的组成和工作原理，数的进制，二进制数的算术运算和逻辑运算，数和字符的编码等。为节省篇幅，本书只对此进行简单复习，不再赘述。

1.1.2 汇编环境介绍

所谓环境，是指在某种操作系统下，利用一种计算机软件（如 MASM、TASM 等），为我们提供一个条件（环境），从而能方便地进行某种计算机语言的编辑和调试。

1. DOS 汇编环境

在 DOS 时代，学习汇编就是学习系统底层编程的代名词，该环境下是 16 位的汇编语言。在 DOS 汇编中，可以采用中断调用功能，以及其他系统内核提供的功能。

2. Win32 汇编环境

Win32 汇编是 Windows 环境下的一种全新的编程语言，使用 Win32 汇编语言是了解操作系统运行细节的最佳方式，它可以当做一种功能强大的开发语言使用。对于习惯于 DOS 汇编环境的程序员来说，会发现曾经学过的内容都被 Windows 封装到内核中了，由于保护模式的存在，程序员无法像在 DOS 环境下那样闯入到系统内核中去。

1.1.3 汇编语言上机过程

汇编语言上机的具体实现过程如下：

- 1) 用编辑程序 (例如 EDIT) 建立 ASM 源文件 (文件名.ASM)。
 - 2) 用汇编程序对 ASM 源文件进行汇编, 产生 OBJ 目标文件 (文件名.OBJ); 若在汇编过程中出现语法错误, 根据错误信息提示 (如位置、类型、说明), 用编辑软件重新调入源程序进行修改。
 - 3) 用链接程序 (例如 LINK) 对目标文件进行链接, 生成 EXE 文件 (文件名.EXE)。
 - 4) 在 DOS 提示符下, 输入 EXE 文件名, 运行程序。
- 汇编语言的上机流程如图 1-2 所示。

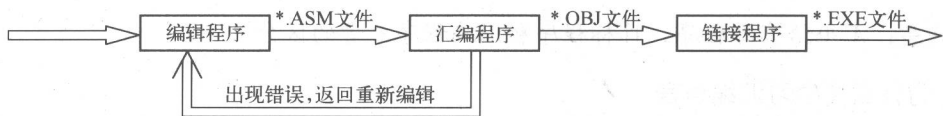


图 1-2 汇编语言上机流程

1.2 微型计算机中数的编码和字符的表示

如何表示数据和字符, 才能让计算机方便、正确、高效地工作呢? 这就是本节所要讨论的内容。

1.2.1 进位计数制

1. 数制

进位计数制简称为数制, 或进位制。在数制中, 涉及 4 个概念: 数位、数码、基数和位权 (权)。

1) 数码: 在数制中, 表示一位数大小的有限个数字符号, 统称为数码。在 N 进制中, 就是自然数 $0, 1, 2, \dots, (N-1)$ 。如八进制中有 8 个数码: $0, 1, 2, \dots, 7$ 。

2) 数位: 数码的序列表示一个数, 数码在序列中的位置称为“数位”。如十进制的个位、十位等。

3) 基数: 某种数制中所拥有基本数码的个数, 即逢几进一。如十进制的基数为 10, 二进制的基数为 2, N 进制的基数为 N 。

4) 位权: 每一个数码在不同的数位上代表的数值不同, 在某一数位上 1 所代表的值称为该数位的位权。如十进制数 128, 其中 1 的位权是 100 (即 10^2), 2 的位权是 10 (即 10^1), 8 的位权是 1 (即 10^0), 所以十进制数中千位、百位、十位、个位上的权可表示为 $10^3, 10^2, 10^1, 10^0$ 。

知道了上述这 4 个概念后, 就可以用一组有序数码, 或者以位权展开多项式的求和形式来表示一个数。例如, $520.919 = 5 \times 10^2 + 2 \times 10^1 + 0 \times 10^0 + 9 \times 10^{-1} + 1 \times 10^{-2} + 9 \times 10^{-3}$ 。

2. 常用的进位计数制

(1) 十进制 (Decimal Notation)

十进制数用 10 个数码 ($0, 1, 2, \dots, 9$) 记数, 基数为 10, 权为 10^n (n 为数位), 运算规则为逢十进一 (加法运算), 借一当十 (减法运算)。十进制数 X , 一般简记为 $(X)_{10}$ 或 X_D , 或 X (即 D 可以省略不写)。利用按权展开的原理, 任何一个十进制数都可以用位权法

表示。

$$\text{例如: } (200)_{10} = 200D = 2 \times 10^2 + 0 \times 10^1 + 0 \times 10^0$$

$$(341.86)_{10} = 341.86D = 3 \times 10^2 + 4 \times 10^1 + 1 \times 10^0 + 8 \times 10^{-1} + 6 \times 10^{-2}$$

(2) 二进制 (Binary Notation)

二进制数用两个数码 (0、1) 记数, 基数为 2, 权为 2^n (n 为数位), 运算规则为逢二进一 (加法运算), 借一当二 (减法运算)。二进制数 X , 一般简记为 $(X)_2$ 或 $X B$ 。利用按权展开的原理, 任何一个二进制数都可以用位权法表示。

$$\text{例如: } (1011)_2 = 1011B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$(11001.01)_2 = 1001.01B = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

(3) 八进制 (Octal Notation)

八进制数用 8 个数码 (0, 1, 2, ..., 7) 记数, 基数为 8, 权为 8^n (n 为数位), 运算规则为逢八进一 (加法运算), 借一当八 (减法运算)。八进制数 X , 一般简记为 $(X)_8$ 或 $X Q$ 。利用按权展开的原理, 任何一个八进制数都可以用位权法表示。

$$\text{例如: } (57)_8 = 57Q = 5 \times 8^1 + 7 \times 8^0$$

$$(115.23)_8 = 115.23Q = 1 \times 8^2 + 1 \times 8^1 + 5 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2}$$

(4) 十六进制 (Hexadecimal Notation)

十六进制数用 16 个数码 (0, 1, 2, ..., 9, A, B, ..., F) 记数, 基数为 16, 权为 16^n (n 为数位), 运算规则为逢十六进一 (加法运算), 借一当十六 (减法运算)。十六进制数 X , 一般简记为 $(X)_{16}$ 或 $X H$ 。在 16 个数码中, 字母 A、B、C、D、E、F 分别对应十进制数中的数字 10、11、12、13、14、15。利用按权展开的原理, 任何一个十六进制数都可以用位权法表示。

$$\text{例如: } (8AC)_{16} = 8ACH = 8 \times 16^2 + A \times 16^1 + C \times 16^0$$

$$(6D.3E)_{16} = 6D.3EH = 6 \times 16^1 + D \times 16^0 + 3 \times 16^{-1} + E \times 16^{-2}$$

上述 4 种进制之间的等值对照关系如表 1-1 所示。

表 1-1 4 种进制之间的等值对照关系

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

1.2.2 进制之间的转换

在实际运算中, 常常涉及不同进制间的相互转换, 它们之间的转换应遵循转换原则。