

高等院校计算机系列教材



GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI



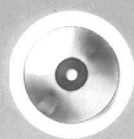
汇编语言程序设计

总主编：陈火旺 主 编：：杨路明 湖南省计算机学会规划教材 中南大学出版社



HBYYCXSJ
HUIBIANYUYAN
CHENGXUSHEJI

高等院校计算机系列教材



GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI



汇编语言程序设计

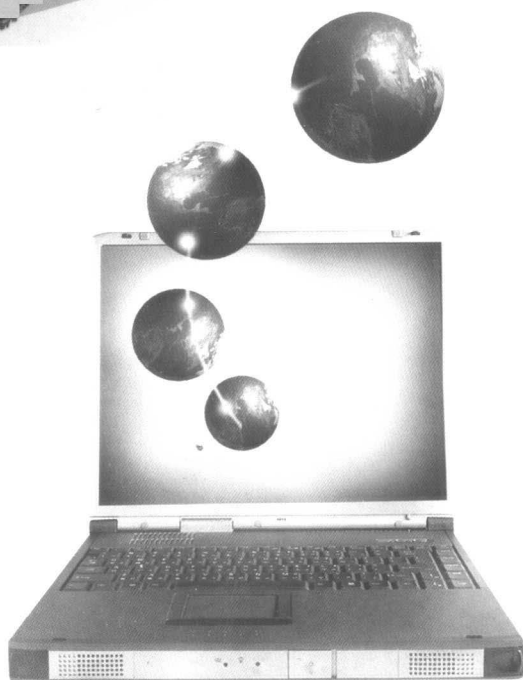
总主编：陈火旺 湖南省计算机学会规划教材 中南大学出版社

主 编：杨路明

副主编：罗庆云 颜拥军 严 晖 喻楚云

编 委：（按姓氏笔画排序）

许纲理 胡宁静 熊曙光



图书在版编目(CIP)数据

汇编语言程序设计/杨路明主编. —长沙:中南大学出版社,
2005. 8

ISBN 7-81105-156-7

I. 汇... II. 杨... III. 汇编语言—程序设计 IV. TP313

中国版本图书馆 CIP 数据核字(2005)第 094384 号

**汇编语言程序设计
(含汇编语言程序设计实验教程)**

杨路明 主编

☐ 责任编辑 陈应征 谭晓萍

☐ 责任印制 文桂武

☐ 出版发行 中南大学出版社

社址:长沙市麓山南路

邮编:410083

发行科电话:0731-8876770

传真:0731-8710482

☐ 印 装 长沙市华中印刷厂

☐ 开 本 787×1092 1/16 ☐ 印张 28.25 ☐ 字数 695 千字 插页 1

☐ 版 次 2005 年 8 月第 1 版 ☐ 2005 年 8 月第 1 次印刷

☐ 书 号 ISBN 7-81105-156-7/TP·011

☐ 定 价 39.00 元

图书出现印装问题,请与出版社调换

高等院校计算机系列教材编委会

总 主 编 陈火旺

执行总主编 孙星明

副 总 主 编 李仁发 陈志刚

编 委 (按姓氏笔画排序)

王志英	刘任任	刘 宏	刘振宇
孙星明	羊四清	阳小华	阳爱民
余绍黔	吴宏斌	张新林	李仁发
李正华	李 军	李勇帆	李 峰
杨路明	沈 岳	肖建华	肖晓丽
陈火旺	陈志刚	罗庆云	金可音
胡志刚	赵 欢	徐建波	殷建平
郭国强	高守平	庾 清	黄国盛
龚德良	傅 明	彭民德	曾碧卿
蒋伟进	鲁荣波	谭骏珊	谭敏生

总 序

21 世纪,人类社会已经步入信息时代,信息产业推动着全球经济的蓬勃发展,改变着人类的联系与交换方式,从某种意义上说,信息革命是人类历史上又一次深刻的社会变革。无疑,在以信息产业为基础的知识经济社会中,计算机科学与技术具有举足轻重的地位。有鉴于此,当今世界各国皆把培养高素质的创新型计算机科学与技术专业人才作为一项重要的战略任务来抓。早在 1984 年,邓小平同志就强调指出:“计算机的普及要从娃娃抓起”,从此开启了中国信息革命的征程。经过 20 多年的努力,我国的计算机教育虽然取得了令人瞩目的成就,但离知识经济社会的要求还有很大的差距。据 2005 年信息产业部的数据显示,我国的信息化人才资源指数仅为 13.43,每年短缺信息化专业人才达 100 万之多。因此,快速培养和造就一大批高素质的计算机与信息人才,乃是我国高等教育所面临的一项严峻挑战。为此,我们必须改革和完善现有计算机与信息技术学科的教学计划和课程体系,优化课程结构,精炼教学内容,拓宽专业基础,强化实践环节,注重学生的知识、能力和综合素质的培养。

为了适应计算机科学与技术学科发展和教育的需要,湖南省计算机学会,参照《中国计算机科学与技术学科教程 2002》,组织了一批长期从事计算机科学与技术专业教学与科研的学者参与编撰了这套由中南大学出版社出版的《高等院校计算机系列教材》,希望在教材中及时反映学科前沿的研究成果与发展趋势,以高水平的科研促进教材建设,以优秀教材促进教学质量的提高。该系列教材具有如下特点:

1. 教材参照《中国计算机科学与技术学科教程 2002》建议的教学大纲、知识领域、知识单元和知识点,结合作者多年教学与科研经验来编写,注重基本理论、基础知识的梳理、推演与挖掘,注意知识的更新,跟踪新技术、新成果的发展,并将之吸收到教材中来,力求开阔学生视野,逐步形成“基础课程精深,专业课程宽新”的格局,努力提高教材质量。

2. 注重理论联系实际,注意能力培养。力图通过案例教学、课堂讨论、课程实验设计与实习,训练学生掌握知识、运用知识分析并解决实际问题的能力以满足学生今后从事科研和就业的需要。

3. 在规范教材编写体例的同时,注重写作风格的灵活性:每册的每个章节包括教学目的、本章小结、思考题与练习题,每门教材都配有 PPT 电子教案,并做到层次分明、逻辑性强、概念清楚、图文并茂、表达准确、可读性强。

这套教材的编写吸纳了广大计算机科学与技术教育工作者多年的教学与科研成果，凝聚了作者们的辛勤劳动，也得到了湖南省各高等院校相关专业领导和专家的大力支持。我相信这套教材的出版，对我国计算机科学与技术专业本科教学质量的提高将有很好的促进作用。

由于编委和作者们水平与时间的限制，教材中难免还有不足之处，恳请广大读者批评指正。

陈火旺

2005 年 7 月

前 言

汇编语言程序设计是电子信息类专业的重要基础课程,是计算机应用开发人员必须接受的最重要的专业基础训练之一,是计算机软、硬件研究的基础,是一个优秀计算机工作者必须掌握的基础知识和基本技能。

本书全面介绍了汇编语言的概念、特性和程序设计方法。全书共有9章,第1章介绍了学习汇编语言程序设计所需的若干计算机基础知识,包括汇编语言的概念,计算机中数据信息的表示方法,微型计算机的组成,微处理器的结构等。第2章介绍了微型机指令系统、指令寻址方式。第3章主要介绍了汇编语言的基本知识。第4章围绕顺序程序设计、分支程序设计以及循环程序设计作了较详细的阐述。第5章对子程序设计技术进行了分析。第6章介绍了宏指令等高级汇编语言技术。第7章主要介绍了计算机输入输出的基本概念和输入输出程序设计的基本方法。第8章对32位机汇编语言程序设计的特点及编程技术作了重点叙述。第9章介绍了Windows环境下汇编语言应用程序设计的实用技术。

本书是作者在多年从事汇编语言程序设计及计算机专业相关课程的教学实践,在多次编写讲义、教材的基础上编写而成的。内容充实,循序渐进,选材上注重系统性、先进性、实用性。全书精选了大量例题,且都作了注释,所有例题均已上机通过,可直接引用。

为配合读者学习本书,我们另编写了一本《汇编语言程序设计上机指导及习题选解》,作为本书的配套参考书,供读者复习和检查学习效果。

本书由杨路明主编,罗庆云、颜拥军、严晖、喻楚云为副主编。杨路明编写第1章;严晖编写第2章、第9章;罗庆云编写第3章;胡宁静编写第4章;喻楚云编写第5章;熊曙光编写第6章;许纲理编写第7章;颜拥军编写第8章。杨路明负责全书的统稿。

由于作者水平所限,加之时间仓促,书中错误和不当之处在所难免,敬请读者批评指正。

作 者

2005年6月于中南大学

目 录

第 1 章 计算机基础知识	(1)
1.1 为什么要用汇编语言编写程序	(1)
1.1.1 计算机语言	(1)
1.1.2 汇编语言的特点	(3)
1.1.3 汇编语言的应用领域	(5)
1.2 计算机中数据信息的表示	(6)
1.2.1 进位计数制	(6)
1.2.2 计算机中带符号数的表示	(8)
1.2.3 计算机中定点数与浮点数的表示	(10)
1.2.4 数的其他编码表示	(10)
1.2.5 基本的数据类型	(12)
1.2.6 逻辑代数简介	(13)
1.3 微型计算机系统	(14)
1.3.1 微型计算机硬件组成	(14)
1.3.2 微型计算机系统组成	(16)
1.4 微处理器	(16)
1.4.1 微处理器的寄存器结构	(17)
1.4.2 存储器的组织管理	(19)
1.5 汇编程序与汇编语言程序开发	(22)
1.5.1 汇编程序简介	(22)
1.5.2 汇编语言程序及其开发	(23)
习 题	(27)
第 2 章 寻址方式和指令系统	(28)
2.1 指令系统概述	(28)
2.1.2 指令系统及指令格式	(28)
2.1.2 8086 指令格式	(28)
2.2 指令寻址方式	(29)
2.2.1 固定寻址方式	(29)
2.2.2 寄存器寻址方式	(29)
2.2.3 立即寻址方式	(29)
2.2.4 数据存储器寻址方式	(30)
2.2.5 与转移地址有关的寻址方式	(33)
2.3 8086 指令系统	(36)

2.3.1	数据传送指令	(36)
2.3.2	算术运算类指令	(41)
2.3.3	位操作指令	(46)
2.3.4	串处理指令	(50)
2.3.5	控制转移指令	(54)
2.3.6	处理机控制指令	(60)
习 题		(61)
第3章 汇编语言		(62)
3.1	概述	(62)
3.1.1	汇编语言的基本规定	(62)
3.1.2	常量	(64)
3.1.3	变量定义及存储器分配	(64)
3.1.4	标号	(68)
3.1.5	表达式	(68)
3.2	汇编语言的伪指令	(72)
3.2.1	段定义伪指令	(72)
3.2.2	过程定义伪指令	(76)
3.2.3	程序开始和结束伪指令	(77)
习 题		(77)
第4章 汇编语言程序设计的基本技术		(80)
4.1	程序设计的基本步骤	(80)
4.2	顺序程序设计	(80)
4.2.1	顺序程序的结构	(80)
4.2.2	汇编语言程序输入输出的实现	(82)
4.2.3	顺序程序示例	(84)
4.3	分支程序设计	(88)
4.3.1	分支程序的基本结构	(88)
4.3.2	简单分支程序示例	(89)
4.3.3	多分支程序设计	(91)
4.3.4	分支程序设计小结	(97)
4.4	循环程序设计	(98)
4.4.1	循环程序的结构	(98)
4.4.2	单重循环程序设计	(99)
4.4.3	多重循环程序设计	(106)
4.4.4	循环程序设计小结	(112)
习 题		(112)
第5章 子程序设计		(115)
5.1	子程序的基本概念	(115)

5.1.1 子程序的基本概念	(115)
5.1.2 子程序的定义与说明	(115)
5.2 子程序的设计方法	(116)
5.2.1 子程序的调用与返回	(117)
5.2.2 现场保护与恢复	(117)
5.2.3 主、子程序的参数传递	(118)
5.2.4 子程序设计举例	(124)
5.3 子程序的嵌套调用与递归调用	(127)
5.3.1 子程序的嵌套调用	(127)
5.3.2 子程序的递归调用	(131)
习 题	(136)
第 6 章 高级汇编语言技术	(138)
6.1 记录和结构	(138)
6.1.1 结构	(138)
6.1.2 记录	(140)
6.2 宏	(141)
6.2.1 宏定义	(141)
6.2.2 宏调用与宏展开	(142)
6.2.3 宏定义与宏调用中的参数	(143)
6.2.4 宏库的使用	(146)
6.3 重复汇编	(148)
6.4 条件汇编	(150)
6.5 程序模块间通信	(152)
6.5.1 程序模块间通信伪指令	(152)
6.5.2 程序模块间通信示例	(153)
6.5.3 多模块程序文件的编译与连接	(154)
习 题	(155)
第 7 章 输入输出与中断	(157)
7.1 输入输出指令与数据传送方式	(157)
7.1.1 输入输出端口地址和输入输出指令	(157)
7.1.2 数据的传送方式	(158)
7.2 中断	(160)
7.2.1 中断的基本概念	(160)
7.2.2 中断处理程序的设计	(163)
7.3 输入输出程序设计	(165)
7.3.1 基本输入输出系统 BIOS 概述	(165)
7.3.2 键盘处理程序	(166)
7.3.3 显示输出程序	(168)

习 题	(176)
第 8 章 32 位机汇编语言编程初步	(177)
8.1 32 位微处理器	(177)
8.1.1 80X86 微处理器的寄存器组	(177)
8.1.2 80X86 保护模式下的存储管理机制	(178)
8.1.3 32 位微处理器扩展寻址方式	(181)
8.2 指令系统扩展	(182)
8.2.1 指令集的扩展	(182)
8.2.2 新增指令	(182)
8.3 32 位汇编语言程序设计基础	(186)
8.3.1 编译工具	(186)
8.3.2 32 位指令的 DOS 程序设计	(187)
8.3.3 Win32 应用编程接口 API	(193)
8.3.4 Win32 编程基础	(194)
习 题	(202)
第 9 章 汇编语言实用程序设计	(203)
9.1 动态链接库	(203)
9.1.1 动态链接库基本知识	(203)
9.1.2 动态链接库的入口函数和导出函数	(204)
9.1.3 动态链接库的使用	(207)
9.2 时钟程序	(210)
9.2.1 计时器	(211)
9.2.2 时钟程序	(211)
9.2.3 编写动态链接库文件	(216)
9.3 磁盘文件管理程序	(221)
9.3.1 磁盘文件管理窗口操作功能及构成	(221)
9.3.2 窗口程序设计	(222)
9.3.3 动态链接库设计	(231)
习 题	(235)
附录 I 8086 指令系统	(236)
附录 II 常用 DOS 系统功能调用 (INT 21H)	(240)
附录 III 常用 ROM - BIOS 功能调用	(243)
附录 IV 动态调试程序 DEBUG 的使用方法	(246)
参考文献	(251)

第1章 计算机基础知识

本章简要地介绍汇编语言的优点及其应用领域。重点介绍了计算机中数据信息的表示方法,包括采用的数制、二进制的特点,定点数和浮点数的表示,数的补码表示,数的其他编码表示等。对微型计算机系统组成,微处理器的结构,特别是微处理器中寄存器性能、用途,以及存储器的组织均作了较详细的阐述。对汇编语言程序作了初步介绍。

电子计算机是人类长期从事计算实践活动的产物,它的发明和发展是20世纪卓越的科学成就之一,也是现代科学技术发展的重要标志。

现代电子计算机大多采用了冯·诺依曼原理,使用程序存储、程序控制技术。计算机的应用遍及科学计算、数据处理、自动控制、智能模拟、辅助设计等众多领域,无论在哪个应用领域,计算机的所有工作都由程序预先安排,计算机每一步的操作都是由程序控制的。实际上,无论是计算机的系统软件还是应用软件,都是用计算机程序设计语言书写的程序。程序的优劣对计算机工作的好坏具有至关重要的影响。

1.1 为什么要用汇编语言编写程序

1.1.1 计算机语言

完成程序设计,自然离不开程序设计语言。不同的问题,可以用不同的程序设计语言来解决,但解决问题的难易程度会各不相同。随着计算机技术的发展,程序设计语言发展非常迅速,新的程序设计语言层出不穷,其功能也越来越强大。从类别看,计算机程序设计语言包括机器语言、汇编语言和高级语言等几类。

1. 机器语言(machine language)

由于物理器件和实现难易的影响,直到目前为止,计算机还只能存放和识别由0和1组成的序列所表示的数据和指令。所谓机器语言,就是指计算机系统能够直接识别的指令的集合,即指令系统。在机器语言中,每条指令都用0和1组成的序列来表示。例如,以下是某计算机的两条机器指令:

加法指令:10000000

减法指令:10010000

可见,机器语言的表现形式是二进制编码,它的每一条语句实际上是二进制形式的指令代码。

由于机器指令与计算机紧密相关,所以,不同种类的计算机所对应的机器指令也就不同,而且它们的指令系统往往相差很大。但对同一系列的计算机来说,指令系统往往具有良好的兼容性。这样,先期开发出来的各类程序才能在新一代的计算机上正常运行。

用机器语言编写程序是早期经过严格训练的专业技术人员的工作,普通的程序员一般难以胜任,而且用机器语言编写的程序可读性较差、出错率较高、维护较困难,又难以直观地反映用计算机解决问题的基本思路。因此,通常不直接用机器语言编写程序。

2. 汇编语言(assembly language)

汇编语言是一种面向机器的程序设计语言,它一般对应着特定的计算机。汇编语言采用一定的助记符号表示机器语言中的指令和有关数据,即用助记符替代二进制形式的机器指令,因此,汇编语言常被称为符号语言。

助记符一般都是表示一个操作的英文字母的缩写,与机器语言相比,便于识别和记忆。例如,上例中的两条指令可用汇编语言描述如下:

加法指令: ADD AX, BX

减法指令: SUB DX, CX

汇编语言包括机器指令集、伪指令集和宏指令集三类。机器指令是指计算机指令系统中的指令。伪指令是由汇编语言需要而设立的指示性命令,它不像机器指令那样包含在指令系统中,其作用是指示系统完成某些操作。宏指令是用机器指令和伪指令定义的可在程序中使用的指令。一条宏指令可对应若干条机器指令,使用宏指令可以使程序简单明了。

用汇编语言编写的程序称为汇编语言程序。但是,计算机硬件不能直接识别和执行汇编语言程序,它必须经过一个称为汇编程序的系统软件翻译成机器语言程序后才能执行。一般称前者为源程序,后者为目标程序。将汇编语言源程序翻译成机器语言目标程序的过程称为汇编,三者的关系如图 1-1 所示。

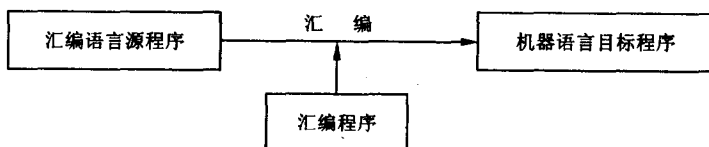


图 1-1 汇编语言源程序、目标程序及汇编程序之间的关系

汇编语言指令和机器语言指令之间具有一一对应的关系。因而,不同型号的计算机系统有着不同的汇编语言,并且程序编写时仍需要对计算机内部结构比较熟悉。在实际计算机应用中,如果程序运行时间要求比较严格,程序与硬件操作联系紧密,人们还是常用汇编语言编写有关程序来解决这些问题。

3. 高级语言(high level language)

高级语言是一种用较为接近自然语言的英文语句和数学表达式表示的编程语言,是面向问题、过程的语言,在 20 世纪 90 年代开始还发展了面向对象的程序设计语言。高级语言具有较大的通用性,一般用高级语言编写的程序较容易移植到不同类型的计算机系统上。

同样,用高级语言编写的程序不能被计算机直接识别和执行。要运行高级语言编写的程序,也要先用一个系统程序将它翻译成计算机能识别的二进制机器指令组成的程序,然后才能为计算机执行。

计算机将高级语言编写的源程序翻译成机器语言目标程序时,常用两种翻译方式:一种是“编译”方式,另一种是“解释”方式。所谓编译方式是,首先把源程序翻译成等价的目标程

序,然后再执行此目标程序。而解释方式是把源程序逐句翻译,翻译一句执行一句,边翻译边执行。

常用的高级语言有 FORTRAN、PASCAL、C、C++、PB、BASIC、Visual Basic、Java 等语言。

1.1.2 汇编语言的特点

1. 用汇编语言编程可以有效地提高编程的质量

前面已谈到,采用高级语言编写的程序需要由编译程序(或解释程序)将它翻译成对应的机器语言程序,机器才能接受。现在,高级语言的编译程序在进行寄存器分配和目标代码生成时,也都有一定程度的优化,但由于所使用的“优化策略”要适应各种不同的情况,所以,这些优化策略只能在宏观上,不可能在微观上、细节上进行优化。这样,通过编译或解释生成的机器语言程序往往比较冗长,占用的存储空间较大,运行速度较慢。

由于汇编语言具有“与机器相关性”的特性,汇编语言程序是与机器语言程序一一对应,用汇编语言编写程序几乎是程序员直接在写执行代码,程序员可以在程序的每个具体细节上进行优化,这也是汇编语言程序执行效率高的原因之一。程序员用汇编语言编写程序时,可充分发挥自己的聪明才智,对机器内部的各种资源进行合理的安排,让它们始终处于最佳的使用状态,这样做的最终效果就是:程序的执行代码短,占存储空间小,执行速度快。

下面是两个程序示例,它们完成相同的功能。

例 1-1 用高级语言 VC++ 编写程序实现:把变量 var1、var2 的内容相加赋给变量 var3,并在屏幕上显示出来。

```
#include "stdafx.h"
#include "stdio.h"

int main(int argc, char * argv[])
{
    int var1, var2, var3; /* 定义三个整型变量 */
    var1 = 1; var2 = 2; /* 变量赋值 */
    var3 = var1 + var2; /* 算术运算并赋值 */
    printf("var3 = %d\n", var3); /* 输出结果 */
    return 0;
}
```

此程序编译后生成的目标文件达到 3.59 KB。

例 1-2 用汇编语言编写程序实现:把变量 var1、var2 的内容相加赋给变量 var3,并在屏幕上显示出来。

```
sseg    segment    stack
        db          256 dup(?)

sseg    ends
dseg    segment

var1    db          1          ;    定义变量 var1
var2    db          2          ;    定义变量 var2
var3    db          ?          ;    定义变量 var3
```

```

string    db        'var3 = $'    ;    定义变量 string
dseg      ends
cseg      segment
          assume     cs: cseg, ds: dseg, ss: sseg
main      proc      far
          push       ds
          sub        ax, ax
          push       ax
          mov        ax, dseg
          mov        ds, ax
          mov        al, var1
          add        al, var2      ;    两数相加
          mov        var3, al     ;    保存运算结果
          lea        dx, string
          mov        ah, 09
          int        21h
          add        var3, 30h
          mov        dl, var3
          mov        ah, 2
          int        21h          ;    显示运算结果
          mov        dl, 0ah
          int        21h
          mov        dl, 0dh
          int        21h
          ret
main      endp
cseg      ends
          end        main        ;    源程序结束

```

此程序汇编后生成的目标文件只有 208 字节。可见在完成相同功能的情况下，汇编语言程序对应的机器代码比高级语言程序生成的代码要小多了。

2. 用汇编语言编程可以充分发挥计算机硬件的功能

高级语言位于计算机系统的上层，用高级语言编程很难对底层的计算机硬件进行有关操作，使程序员无法直接利用机器硬件系统的许多特性，例如寄存器、状态标志位以及一些特殊指令等，影响许多程序设计技巧的发挥。用汇编语言编程可以充分利用机器硬件的功能，进行涉及计算机系统底层软件的开发。如操作系统的核心模块、设备驱动程序、中断处理程序编写；完成系统加密、解密处理；实现对计算机病毒的分析和防治处理等。

3. 用汇编语言编程可以进行输入输出接口开发

在微型机系统、实时控制系统、智能化仪器仪表和嵌入式系统的开发应用中，常常需要将主机与外设相连，需要开发输入输出接口。目前在这个领域高级语言尚无法胜任，利用汇编语

言可以很方便地编写出接口硬件初始化、接口硬件调用以及满足各种功能要求的接口程序。

4. 程序的编写、调试和跟踪比较复杂

汇编语言是一种面向机器的语言，汇编指令也同机器指令一样具有功能单一、具体的特点。要想完成某件工作，就必须安排计算机的每步工作。因此，要求程序员熟悉计算机的内部结构，特别是中央处理器和存储器的结构。此外，还应熟悉机器中与编程有关的其他部分的结构，例如，指令系统、中断系统、输入输出系统、操作系统的结构等。

在编写汇编语言程序时，还要考虑机器资源的限制、汇编指令的细节和限制等等。这就使得编写汇编语言程序比较繁琐、复杂。

在通常情况下，调试汇编语言程序也比调试高级语言程序困难，其主要原因如下：

(1) 汇编语言指令涉及机器资源的细节，在调试过程中，要清楚每个资源的变化情况。

(2) 程序员在编写汇编语言程序时，为了提高资源的利用率，可以使用各种实现技巧，而这些技巧可能破坏程序的可读性。这样，在调试过程中，除了要知道每条指令的执行功能外，还要清楚它在整个解题过程中的作用。

(3) 汇编语言程序要用到大量的、各类转移指令，这些跳转指令大大地增加了调试程序的难度。

(4) 调试工具落后，高级语言程序可以在源程序级进行符号跟踪，而汇编语言程序只能跟踪机器指令。

正因为汇编语言程序的编写、调试和跟踪比较复杂，非专门人员难以掌握。在计算机应用领域，往往利用汇编语言的这个特点，在关键系统的部位采用汇编语言编程，从技术上有效地实现对软硬件模块的保护。

1.1.3 汇编语言的应用领域

汇编语言的优点是“面向机器”、“占存储空间小”、“执行的高效率”和“可以直接驱动硬件”，但其可移植性差、调试难度大。因此，是否采用汇编语言编写程序，要根据实际的应用环境，在软件的开发时间及软件的质量方面进行权衡和决策。

对于以下应用情况，适于采用汇编语言编程：

(1) 对运行时间、存储容量要求较高的程序。例如，操作系统的内核程序、硬件驱动程序。

(2) 与硬件资源密切相关的软件开发，需要程序直接控制硬件的场合。例如，输入输出接口电路的初始化程序、外部设备的低层驱动程序等。

(3) 计算机系统性能的瓶颈处，或频繁被使用的子程序或程序段，或需要提高大型软件性能的场合。

(4) 从信息安全的角度考虑，计算机系统低层软件分析，信息的加密与解密，计算机病毒的分析与防治。

(5) 计算机用于实时控制，且要求执行效率高、反应快的领域。例如，工业控制系统、实时系统的软件、智能仪器仪表的控制程序，宜用汇编语言编写。

(6) 受存储容量限制的计算机应用领域，例如，家用电器的计算机控制功能等。

(7) 在没有合适的高级语言开发环境的场合。

总之，在计算机技术迅速发展的今天，汇编语言程序设计技术仍然占有相当重要的地

位,它是一个优秀计算机工作者必须掌握的基础知识和基本技能。

1.2 计算机中数据信息的表示

由于汇编语言是面向机器的语言,程序员在用汇编语言进行程序设计时,需要直接使用计算机的各种资源,这样就需要对数据在机器中的表示有一个清晰的认识。

1.2.1 进位计数制

在日常生活中,可遇到各种进位计数制,如十进制、十二进制等。要表示一个数,就要选择适当的数字符号和规定其组合规律,也就是确定所选用的进位计数制。

每种进位计数制都有其基本的因素:基数和位权。基数表示了进位制所具有的数字符号的个数以及进位的规律;位权则表示同一个数字符号在数的不同的数位上有不同的数值。

1. 十进制

十进制是我们最熟悉的一种数据表示形式,它具有两个主要特点:

(1) 使用 10 个不同的数字符号,即 0、1、2、3、4、5、6、7、8、9,所以它的基数为十,故称为十进制。

(2) 它的计算规律是“逢十进一”。

一个十进制数各数位的位权为:

从小数点向左,位权依次是 $1(10^0)$ 、 $10(10^1)$ 、 $100(10^2)$ 、 $1000(10^3)$ 、 $\dots$;从小数点向右依次是 $0.1(10^{-1})$ 、 $0.01(10^{-2})$ 、 $0.001(10^{-3})$ 、 $\dots$ 。高一位的权是低一位的 10 倍。

在汇编语言程序中经常用十进制来表示数据。书写时为了区别起见,在数据后面紧跟一个字母 D(或省略 D)。

2. 二进制

在计算机内部,信息的存储和处理都采用二进制。与十进制类似,它也有两个特点:

(1) 它使用两个不同的数字符号,即 0、1,所以它的基数为 2,故称为二进制。

(2) 它的计算规律是“逢二进一”。

一个二进制数各数位的位权为:

从小数点向左,位权依次是 $1(2^0)$ 、 $2(2^1)$ 、 $4(2^2)$ 、 $8(2^3)$ 、 $\dots$;从小数点向右依次是 $0.5(2^{-1})$ 、 $0.25(2^{-2})$ 、 $0.125(2^{-3})$ 、 $\dots$ 。高一位的权是低一位的 2 倍。

同一个数码在不同的数位所代表的值不同。例如:

$$(1011.1)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$$

在程序中书写二进制时,为了区别起见,在数据后面紧跟一个字母 B。

二进制的一般表示形式为:

$$b_{n-1} \dots b_1 b_0 B, (b_i = 0, 1)$$

其代表的数值(十进制)为: $b_{n-1} \times 2^{n-1} + \dots + b_1 \times 2^1 + b_0 \times 2^0$ 。

各种信息在计算机内均表示为二进制形式,所以信息的最小单位就是二进制的“位”(Bit),在实际使用中,常用的单位还有“字节”(Byte),一个字节的长度为 8 位。计算机中完成一次基本处理的二进制数据位数称为计算机的字长,微型机的字长有 8 位、16 位、32 位、64 位等多种。