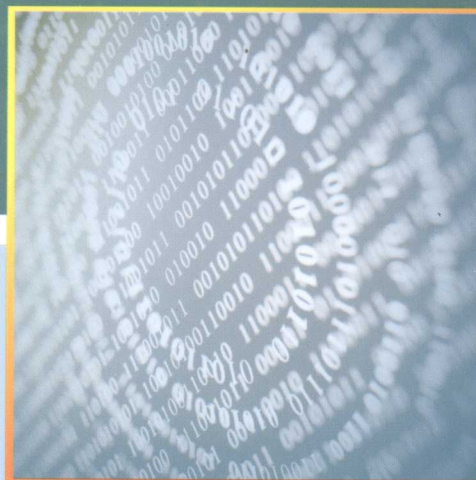


普通高等学校计算机科学与技术专业规划教材

汇编语言程序设计

ASSEMBLY LANGUAGE PROGRAMMING

白小明 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等学校计算机科学与技术专业规划教材

汇编语言程序设计

主 编 白小明
副主编 徐 苏 娄 芳
参 编 陈 萌

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书根据 IEEE/ACM 推出的 CC2005 和我国教育部高等学校计算机科学与技术教学指导委员会 2006 年发布的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》编写而成，内容涵盖了知识领域 CS-AR3 汇编级机器组织的知识点。

本书以 Intel 80x86 指令系统和汇编语言开发工具 MASM 6.11 与 MASM 32 为主体，在 PC 的 MS-DOS 和 Windows 环境中，较为系统地介绍了汇编语言程序设计所需要的指令、语法以及调试工具。第 1~5 章是 80x86 汇编语言的基本内容，包括汇编语言基础知识、寻址方式、主要汇编指令、伪指令、源程序格式、分支结构程序、循环结构程序、子程序和宏的程序设计技巧；第 6~8 章介绍了输入/输出程序的设计，包括输入/输出的基本概念，查询方式的 I/O 程序设计，中断处理程序设计，键盘、鼠标和视频的操作，磁盘文件存取的相关程序设计；第 9~10 章将内容扩展到 32 位汇编语言、多媒体指令和浮点编程等应用方面以及 Windows 窗口应用程序的开发。

本书适合作为高等院校计算机科学与技术及相关专业的教材，也可作为相关专业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/白小明主编. —北京：中国铁道出版社，2009.6

（普通高等学校计算机科学与技术专业规划教材）

ISBN 978-7-113-10101-5

I. 汇… II. 白… 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 109323 号

书 名：汇编语言程序设计

作 者：白小明 主编

策划编辑：严晓舟 周海燕

责任编辑：秦绪好

编辑部电话：（010）63583215

编辑助理：包 宁

封面设计：付 巍

封面制作：白 雪

版式设计：郑少云

责任印制：李 佳

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号 邮政编码：100054）

印 刷：中国铁道出版社印刷厂

版 次：2009 年 8 月第 1 版 2009 年 8 月第 1 次印刷

开 本：787mm×1092mm 1/16 印张：22 字数：536 千

印 数：5 000 册

书 号：ISBN 978-7-113-10101-5/TP·3322

定 价：35.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签，无标签者不得销售

凡购买铁道版的图书，如有缺页、倒页、脱页者，请与本社计算机图书批销部调换。

普通高等学校计算机科学与技术专业规划教材

主 任：蒋宗礼（北京工业大学）

副主任：王志英（国防科技大学）

杨 波（济南大学）

委 员：（按姓氏音序排列）

常会友（中山大学）

陈 明（中国石油大学）

陈志国（河南大学）

胡 亮（吉林大学）

姜守旭（哈尔滨工业大学）

刘腾红（中南财经政法大学）

王国仁（东北大学）

吴 跃（电子科技大学）

岳丽华（中国科技大学）

陈俊杰（太原理工大学）

陈笑蓉（贵州大学）

顾乃杰（中国科技大学）

黄国兴（华东师范大学）

李仲麟（华南理工大学）

罗军舟（东南大学）

王命延（南昌大学）

袁晓洁（南开大学）

张 莉（北京航空航天大学）

本书责任编辑：王命延（南昌大学）

计算学科虽然是一门年轻的学科，但它已经成为一门基础技术学科，在各个学科发展中扮演着重要的角色，并使得社会产生了对计算机科学与技术专业人才的巨大需求。目前，计算机科学与技术专业已成为我国理工专业中规模最大的专业，为高等教育发展做出了巨大贡献。近些年来，随着国家信息化建设的推进，作为核心技术的计算机技术，更是占有重要的地位。信息化建设，不仅需要更先进、更便于使用的先进计算技术，同时也需要大批的建设人才。瞄准社会需求准确定位，培养计算机人才，是计算机科学与技术专业及其相关专业的历史使命，也是实现专业教育从劳动就业供给导向型向劳动就业需求导向型转变的关键，从而也就成为提高高等教育质量的关键。

教材在人才培养中占有重要地位，承担着“重要的责任”，这确定了其高质量的基本要求。社会对计算机专业人才需求的多样性和特色，决定了教材建设的针对性，从而也造就了百花齐放、百家争鸣的局面。

关于建设高质量的教材，教育部在提高本科教育质量的文件中均提出了明确要求。教高〔2005〕1号（2005年1月7日）文件指出，“加强教材建设，确保高质量教材进课堂。要大力锤炼精品教材，并把精品教材作为教材选用的主要目标。”“要健全、完善教材评审、评价和选用机制，严把教材质量关。”为了更好地落实教育部的这些要求，我们按照教育部高等学校计算机科学与技术教学指导委员会发布的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》所构建的计算机科学与技术专业本科教育的要求，组织了这套教材。

作为优秀教材的基础，我们首先坚持高标准，以对教育负责的精神去鼓励、发现、动员、选拔优秀作者，并且有意识地培育优秀作者。优秀作者保证了“理论准确到位，既有然，更有所以然；实践要求到位、指导到位”等要求的实现。

其次是按照人才培养的需要适当强调学科形态内容。粗略地讲，计算机科学的根本问题是“什么能被有效地自动计算”，科学型人才强调学科抽象和理论形态的内容；计算机系统工程的根本问题应该是“如何低成本、高效地实现自动计算”，工程型人才强调学科抽象和设计形态的内容；计算机应用的根本问题是“如何方便、有效地利用计算机系统进行计算”，应用型人才的培养偏重于技术层面的内容，强调学科设计形态的内容，在进一步开发基本计算机系统应用的层面上体现学科技术为主的特征。教材针对不同类型人才的培养，在满足基本知识要求的前提下，强调不同形态的内容。

第三是重视知识的载体作用，促进能力培养。在教材内容的组织上，体现大学教育的学科性和专业性特征，参考《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》示例性课程大纲，覆盖其要求的基本知识单元。叙述上力争引导读者进行深入分析，努力使读

者在知其然的基础上，探究其所以然。通过加强对练习和实践的引导，进一步培养学生的能力，促使相应课程在专业教育总目标的实现中发挥作用。

第四是瞄准教学需要，提供更多支持。近些年来，随着计算机技术、网络技术等在教学上的应用，教学手段、教学方式不断丰富，教材的立体化建设对丰富教学资源发挥了重要作用。通常，除主教材外，还要配套教学参考书、实验指导书、电子讲稿、网站等。

第五是面向主要读者，强调教材的写作特征，努力做到叙述清晰易懂，语言流畅，深入浅出，有吸引力而不晦涩；追求描述的准确性，强调用词和描述的一致性，语言表达的清晰性和叙述的完整性；分散难点，循序渐进，防止多难点、多新概念的局部堆积。

我们相信，这套教材一定能够在培养社会需要的计算机专业人才上发挥重要作用，希望大家广为选用，并在使用中提出宝贵建议，使其内容不断丰富。

普通高等学校计算机科学与技术专业规划教材编审委员会

2008年1月

“汇编语言程序设计”是继“高级语言程序设计”之后的又一门计算机语言程序设计课程，它讲解的是面向机器硬件的低级语言。汇编语言能够为程序员提供最直接操作机器硬件系统的途径，利用它可以编写出在“时间”和“空间”两个方面最具效率的程序。

“汇编语言程序设计”是计算机各专业的一门重要基础课程，是必修的核心课程之一，是“操作系统”和“微机原理与接口技术”等其他核心课程必要的先修课。该课程一方面可加强学生的高级语言编程能力；另一方面配合“计算机组成原理”和“微机原理及接口技术”课程，从软件角度理解计算机工作原理；同时，还可作为自动控制等与硬件相关应用领域的程序设计基础，以及为“操作系统”、“嵌入式系统”等课程提供基础知识。

★ 本书的主要特色

在多年的汇编语言程序设计课程教学过程中，我们使用过的教材有：沈美明编著的“IBM-PC 汇编语言程序设计”、周明德主编的“80x86 的结构与汇编语言程序设计”、张怀莲主编的“宏汇编语言程序设计”、钱晓捷主编的“汇编语言程序设计”等。这些教材都是很优秀的教材，被很多高校采用。经过二十几年的发展，Intel 处理器技术和汇编语言程序设计方法有了全新的进展。为了跟踪新技术，给读者构造合理的学习线索，需要真正地遵循循序渐进的原则。一方面，我们需要打破传统的章节划分，对课程的内容进行重组、分割和补充；另一方面，根据多年从事本课程教学的经验，我们觉得对本课程的教学思路需要做一些调整。本书在编写上有如下特色：

第一，本书根据 IEEE/ACM 推出的 CC2005 和我国教育部高等学校计算机科学与技术教学指导委员会 2006 年发布的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》编写而成，内容涵盖了知识领域 CS-AR3 汇编级机器组织的知识点，基本涵盖了当前汇编语言的内容，适用范围较为广泛。

第二，本书的编写目的是使汇编语言的编程和学习变得更轻松、更深入、更有趣。在教学内容编排上，由浅入深，循序渐进，对知识点的逻辑结构进行了精心设计安排。

第三，本书主要通过例证来教学。本书在第 1 章就给出了一个完整的汇编语言程序，并且在学生能够理解的层次上，仔细考察了程序的各个部分。随后的章节包含了许多汇编语言代码的例子，同时，对一些新的或者难以理解的概念给出了恰当的解释。本书使用了大量的图表和例子，给出许多“指令执行前”和“指令执行后”的例子来讲解指令，以及一些演示调试程序使用的例子。这些例子可以帮助学生深入了解计算机内部的工作。同时，在每一章节后配以典型的、多样的练习题，汇编语言实验指导部分配有实用性的实验题目，从而引导学生饶有兴趣地学习理论知识，通过完成练习题目，能动地完成实验，在潜移默化中巩固和加深对知识点的理解和掌握。

★ 本书内容的组织

为了适应任何层次的编程，程序员必须了解在机器层的某些基本原理，它们在大多数的计算机体系结构中都要用到。本书将涉及以下基本概念：

- 存储器地址、CPU 寄存器及其使用。

- 计算机中数值型格式的数据和字符串的表示。
- 指令格式。
- 数据的存储方式与寻址方式。
- 指令集及其分类（数据操作、控制、输入/输出）。
- 子程序调用和返回机制。
- 中断处理。
- 汇编语言和机器语言编程基础。

本书共分 10 章。第 1 章介绍汇编语言的基本概念、数据在计算机中的表示、80x86 的寄存器结构、存储器和堆栈的结构；第 2 章以 80x86 为背景，讲解了指令操作数的寻址方式，数据处理指令（包括传送、算术运算、逻辑运算指令）；第 3 章介绍了汇编语言的常用伪指令、汇编语言程序格式和如何利用 DOS 提供的功能调用实现基本输入/输出操作；第 4 章介绍了汇编语言程序的基本结构，转移地址寻址方式以及转移指令、循环指令、串操作指令的格式和功能，通过实例讲解如何利用汇编语言实现程序的分支和循环结构；第 5 章详细讲解了子程序调用与返回指令，子程序的定义及参数传递，宏定义和宏调用等内容；第 1~5 章可作为初学者学习汇编语言程序设计的基本内容。第 6 章以 PC 及其兼容机为硬件平台，介绍输入/输出和中断等概念，讲解如何利用汇编语言编写 I/O 程序和中断处理程序；第 7 章介绍了键盘、鼠标和视频显示的工作原理，讨论了利用 BIOS 中断调用实现键盘、鼠标操作和视频显示程序的设计；第 8 章介绍了磁盘系统的基本知识，讲解了以文件代号方式实现磁盘文件存取的相关程序设计；第 6~8 章的内容可作为汇编语言的提高部分。第 9~10 章将内容扩展到 32 位汇编语言、多媒体指令和浮点指令编程等应用方面，还简要介绍了 Windows 汇编程序的开发过程及 Windows 应用程序的设计；这两章的内容可作为读者了解高档微处理器、保护方式编程和 Windows 汇编程序设计技术的参考书。汇编语言实验指导部分列举了大量实验题目。最后的附录 A 部分介绍汇编语言的调试工具，附录 B 部分给出了 DOS 系统功能调用和 BIOS 中断调用内容，应在上机实验前先阅读了解附录部分内容。

★ 教学学时安排

本书建议总学时为 48~64 学时，各高校按照计算机专业课程体系中的课程设置和讲授内容的不同可以灵活调整。

本书由白小明担任主编，徐苏、娄芳任副主编，陈萌参编，由白小明最后统一定稿。本书从编写大纲的制定到定稿的全过程得到南昌大学王命延教授的热情关心和大力支持，承蒙他审阅了全书，在此表示衷心感谢！本书在编写过程中还得到了同事程从从教授、邱桃荣教授和余松森博士的帮助，在此一并表示感谢！

限于编者的能力和水平，书中不妥与疏漏之处在所难免，恳请同行和读者批评指正。

编 者

2009 年 5 月

第 1 章 汇编语言入门	1
1.1 认识汇编语言	1
1.1.1 机器语言	1
1.1.2 汇编语言	2
1.1.3 一个显示字符串的汇编语言程序	3
1.2 数据在计算机中的表示	4
1.2.1 数值数据在计算机中的表示	4
1.2.2 BCD 码表示	7
1.2.3 字符编码	8
1.3 80x86 微处理器	11
1.3.1 80x86 微处理器的演化	11
1.3.2 80x86 微处理器的组成	13
1.3.3 80x86 微处理器的编程结构	14
1.4 存储器与堆栈	16
1.4.1 存储器	17
1.4.2 实模式下存储器的组织	18
1.4.3 保护模式下存储器的组织	21
1.4.4 堆栈的结构与数据存取	22
本章小结	24
习题	24
第 2 章 寻址方式和指令	26
2.1 80x86 指令系统概述	26
2.1.1 80x86 的指令种类	26
2.1.2 80x86 的指令格式	29
2.2 80x86 指令操作数的寻址	31
2.2.1 80x86 指令的操作数类型	31
2.2.2 立即数寻址	32
2.2.3 寄存器操作数的寻址	33
2.2.4 存储器操作数的寻址	34
2.3 80x86 的几类基本指令	39
2.3.1 数据传送类指令	39
2.3.2 算术运算类指令	46
2.3.3 逻辑运算类指令	52
2.3.4 处理器控制指令	59
本章小结	60

习题	61
第3章 汇编语言的要素	65
3.1 伪指令	65
3.1.1 段定义伪指令	65
3.1.2 源程序开始和结束伪指令	71
3.1.3 数据定义及存储器分配伪指令	72
3.1.4 其他伪指令	75
3.2 汇编语言语句格式	78
3.2.1 名字项	78
3.2.2 操作项	79
3.2.3 操作数项	79
3.2.4 注释项	83
3.2.5 汇编语言源程序的格式	83
3.3 常用的DOS系统功能调用	85
3.3.1 概述	85
3.3.2 常用的输入/输出系统功能调用	85
本章小结	90
习题	90
第4章 程序逻辑与控制	93
4.1 程序控制结构	93
4.1.1 分支程序结构	93
4.1.2 循环程序结构	94
4.1.3 子程序结构	95
4.2 分支结构程序设计	95
4.2.1 无条件转移指令	96
4.2.2 条件转移指令	98
4.2.3 分支程序设计方法	102
4.3 循环结构程序设计	107
4.3.1 循环指令	107
4.3.2 单重循环程序设计	108
4.3.3 多重循环程序设计	113
4.4 串操作	115
4.4.1 串操作指令	115
4.4.2 字符串程序处理程序举例	121
本章小结	123
习题	123
第5章 子程序和宏	125
5.1 子程序	125
5.1.1 子程序调用与返回指令	125

5.1.2	子程序设计的一般方法	130
5.1.3	子程序设计举例	134
5.2	宏	139
5.2.1	宏定义与宏调用	140
5.2.2	宏定义中的参数	142
5.2.3	宏操作符	143
5.2.4	嵌套的宏	144
5.2.5	与宏相关的伪指令	146
5.3	重复汇编和条件汇编	148
5.3.1	重复汇编	148
5.3.2	条件汇编	149
	本章小结	150
	习题	151
第 6 章	输入/输出与中断	153
6.1	输入/输出的基本概念	153
6.1.1	I/O 地址与输入/输出指令	153
6.1.2	数据传送控制方式	156
6.1.3	存取 RT/CMOS RAM	157
6.2	查询方式数据传送	159
6.2.1	查询方式	159
6.2.2	查询方式输入/输出程序设计	160
6.3	80x86 的中断系统	162
6.3.1	80x86 的中断分类	163
6.3.2	中断向量表	164
6.3.3	中断处理过程	167
6.3.4	中断优先级和中断嵌套	168
6.4	中断处理程序设计	169
6.4.1	设置和获取中断向量	169
6.4.2	中断处理程序的设计	171
6.4.3	BIOS 中断调用	177
	本章小结	179
	习题	180
第 7 章	键盘、鼠标和视频的操作	181
7.1	键盘输入	181
7.1.1	BIOS 键盘数据区	181
7.1.2	键盘输入的 INT 16H 操作	185
7.2	鼠标操作	187
7.2.1	基本的鼠标操作	187
7.2.2	鼠标操作程序设计举例	189
7.3	视频的设置	192

7.3.1 文本显示模式	192
7.3.2 彩色文本模式的 BIOS 中断调用	194
7.3.3 图形显示模式	196
本章小结	202
习题	203
第 8 章 磁盘文件存取	204
8.1 磁盘的记录方式	204
8.1.1 磁盘记录信息的地址	204
8.1.2 磁盘系统区和数据区	206
8.1.3 磁盘目录及文件分配表	207
8.2 磁盘存取方式——文件代号	208
8.2.1 文件代号式磁盘存取的基本概念	209
8.2.2 常用的磁盘文件管理 DOS 功能调用	211
8.2.3 写磁盘文件	213
8.2.4 读磁盘文件	216
8.2.5 移动读/写指针	220
8.3 BIOS 磁盘存取中断调用和编程	224
8.3.1 基本的 BIOS 磁盘操作	224
8.3.2 BIOS 磁盘操作程序设计举例	226
本章小结	229
习题	229
第 9 章 汇编语言扩展	231
9.1 32 位指令及其编程	231
9.1.1 32 位处理器寄存器结构	231
9.1.2 32 位处理器指令系统	234
9.1.3 32 位指令的程序设计	236
9.2 多媒体指令	239
9.2.1 MMX 指令	239
9.2.2 SSE 指令	246
9.2.3 SSE 指令集的发展	253
9.3 浮点运算指令及编程	255
9.3.1 协处理器的内部结构	255
9.3.2 协处理器的指令系统	258
9.3.3 浮点运算指令编程举例	262
本章小结	264
习题	265
第 10 章 Win32 汇编	267
10.1 Win32 汇编的环境和基础	267
10.1.1 Win32 汇编的编程环境	267

10.1.2	Win32 汇编程序的结构	272
10.1.3	API 函数的使用	274
10.2	Win32 汇编语言的基本语法	277
10.2.1	标号和变量	277
10.2.2	子程序设计	278
10.2.3	高级语法	280
10.3	Win32 汇编程序设计	283
10.3.1	Win32 窗口程序设计	283
10.3.2	菜单和加速键程序设计	288
10.3.3	工具栏和状态栏的使用	294
10.3.4	图形界面的操作	296
	本章小结	299
	习题	300
	汇编语言实验指导	301
	实验 1 初级程序的编写与调试实验	301
	实验 2 汇编语言程序上机过程	304
	实验 3 分支程序实验	305
	实验 4 循环程序实验	307
	实验 5 子程序实验	308
	实验 6 字符处理程序实验	311
	实验 7 键盘和窗口程序	314
	实验 8 显示器与磁盘文件 I/O 程序设计	315
附录 A	汇编语言编程和调试工具	319
A.1	宏汇编 MASM	319
A.2	调试工具	324
附录 B	DOS 系统功能调用和 BIOS 中断调用	327
	参考文献	335

第1章

汇编语言入门

汇编语言是一门与机器硬件紧密相关的程序设计语言,其核心是机器处理器的指令系统。不同处理器的指令系统是不尽相同的,因此不同机器的汇编语言也就不同。通过学习汇编语言,可以使我们了解相关计算机的硬件结构,深入理解计算机执行指令的过程等。

本章首先介绍机器语言和汇编语言,并通过一个汇编语言程序的例子让读者对汇编语言有一个感性的认识;然后介绍数值数据及非数值数据在计算机中的表示方法;最后介绍80x86微处理器及其存储器的组织。

1.1 认识汇编语言

计算机语言是人与计算机进行交流的工具,通过计算机语言,我们可以告知或命令计算机需要做什么工作,以及完成工作的过程、步骤等。人们使用计算机语言编写计算机程序,因此计算机语言又称为程序设计语言。

1.1.1 机器语言

从一般使用者的角度来看,计算机系统是由硬件和软件组成的。硬件即计算机系统中所有实体部件和设备的总称,如CPU、主板、内存、电源、机箱、显示器、鼠标和键盘等;而软件则是用计算机语言编写的程序,程序可以看成是具有一定逻辑关系的指令的集合,人们正是通过让机器执行一条条指令来完成相应功能的。但是,任何一台机器所执行的指令都是按照一定的语法、语义进行定义的,如指令的格式、指令的编码、指令的操作、指令操作的对象等。一台机器的指令及语法、语义的集合就构成了这台机器的计算机语言。

计算机语言按照与硬件相关程度由高到低分为机器语言、汇编语言和高级语言。机器语言属于硬件机器级语言,是一种用二进制代码表示的能够被计算机硬件直接识别和执行的语言。早期的计算机只提供了机器语言,程序员只能使用机器语言编写程序。由于机器语言与具体机器的硬件紧密相关,因此要求程序员对机器的硬件组成与结构十分清楚,如CPU的内部结构、CPU的寄存器组织、CPU对存储器的组织与操作等。以下通过一个例子来说明如何使用机器语言编写程序。

【例 1.1】假设某机器所有指令的代码长度为8位二进制,其格式如下:

OP	R ₁	R ₂
----	----------------	----------------

其中, OP 表示该机器指令的操作码, 定义如下:

OP	含义
0000	将源寄存器的内容传送到目的寄存器
0001	将目的寄存器的内容与源寄存器的内容相加, 结果送目的寄存器
0010	将目的寄存器的内容减去源寄存器的内容, 结果送目的寄存器
0011	将目的寄存器的内容与源寄存器的内容相乘, 结果送目的寄存器
0100	将目的寄存器的内容除以源寄存器的内容, 结果送目的寄存器

...

R_2 、 R_1 分别为源寄存器和目的寄存器, 定义如下:

R_1/R_2	含义
00	寄存器 A
01	寄存器 B
10	寄存器 C
11	寄存器 D

以下一段机器语言程序实现 x^2+y^2 运算, 其中 x 、 y 已分别存放在寄存器 A、B 中, 计算结果存入寄存器 C 中。

机器指令	含义
00110000	将寄存器 A 的内容与寄存器 A 的内容相乘, 结果送寄存器 A (计算 x^2)
00110101	将寄存器 B 的内容与寄存器 B 的内容相乘, 结果送寄存器 B (计算 y^2)
00010001	将寄存器 A 的内容与寄存器 B 的内容相加, 结果送寄存器 A (计算 x^2+y^2)
00001000	将寄存器 A 的内容传送到寄存器 C

从上面的例子可以看出, 机器语言程序就是由一条条机器指令二进制代码组成, 这些机器指令二进制代码可以由机器硬件直接识别和执行。

早期人们所编写的机器语言程序是通过一种纸带穿孔机设备记录到穿孔纸带上的, 图 1-1 所示为将以上一段程序记录在穿孔纸带上的示意图, 其中每一行代表一条机器指令, 穿孔处为“1”, 未穿孔处为“0”。当需要执行程序时, 只需使用纸带阅读机将记录在纸带上的程序读入到计算机的内存中启动执行即可。后来, 随着键盘的出现, 人们可以将事先编写好的机器语言程序通过键盘输入到机器中存储并执行。

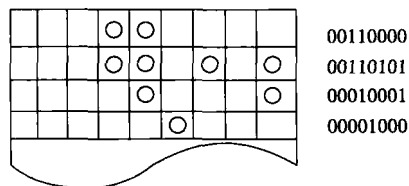


图 1-1 用穿孔纸带记录机器语言程序

机器语言编写的程序执行效率是最高的, 但它对程序员的要求也很高, 一方面要求程序员对机器的硬件非常了解, 另一方面程序员需要记忆大量的二进制指令代码。一台实际机器的指令往往有几十至上百条, 不同指令的代码长度、寻址方式等有可能各不相同, 很不容易记忆。因此, 使用机器语言编写复杂的程序往往是很困难的, 需要花费大量的时间和精力。鉴于此, 人们开始研究一种既与机器硬件相关又便于记忆的计算机语言, 这就是汇编语言。

1.1.2 汇编语言

汇编语言 (assembly language) 是面向机器的语言, 是利用计算机所有硬件特性并能直接控制硬件的语言。汇编语言是一种采用助记符表示的程序设计语言, 它的指令和机器语言

的指令在很大程度上是一一对应的，其指令格式及语法、语义等和机器语言也基本一致。由于汇编语言使用一些表示指令操作的英文单词缩写来代替二进制序列，因此便于程序员记忆和编程。例 1.2 是例 1.1 对应的一段汇编语言程序。

【例 1.2】以下一段汇编语言程序实现 x^2+y^2 运算，其中 x 、 y 已分别存放在寄存器中 A、B，计算结果存入寄存器 C 中。

```
MUL A,A ;将寄存器 A 的内容与寄存器 A 的内容相乘,结果送寄存器 A(计算  $x^2$ )
MUL B,B ;将寄存器 B 的内容与寄存器 B 的内容相乘,结果送寄存器 B(计算  $y^2$ )
ADD A,B ;将寄存器 A 的内容与寄存器 B 的内容相加,结果送寄存器 A(计算  $x^2+y^2$ )
MOV C,A ;将寄存器 A 的内容传送到寄存器 C
```

其中,4 条汇编指令 MUL A,A/MUL B,B/ADD A,B/MOV C,A 分别代替了 4 条机器指令 00110000/00110101/00010001/00001000。MUL、ADD 和 MOV 分别表示“乘”(multiply)、“加”(adding)和“传送”(move)。显然,这种使用助记符表示的汇编指令远比单调的二进制序列好记,且可读性好。

当然,不同机器的汇编指令所使用的助记符有可能不同。例如,80x86 CPU 实现寄存器与寄存器之间以及寄存器与存储单元之间的数据传送均使用 MOV 助记符,而 DEC 公司的 PDP/VAX 机器的数据传送使用 LOD(load)和 STR(store)等。不同机器对寄存器的表示也有所不同。例如,80x86 CPU 的寄存器分别使用 AX、BX、CX 等表示,而 PDP/VAX 机器的寄存器分别使用 R₁、R₂、R₃ 等表示。

助记符表示的汇编指令虽然方便了程序员,但不要忘记:机器只读得懂“0”和“1”,读不懂这些英文字母,不能将这些汇编指令直接交给机器执行。因此人一机之间需要一个“翻译”,将程序中汇编语言指令翻译成机器语言指令,这一“翻译”就是“汇编器”(assembler)或“汇编程序”。也就是说,使用汇编语言指令编写的程序称为“源程序”,源程序通过汇编器的翻译转换成机器语言程序后,才能由机器识别和执行。

1.1.3 一个显示字符串的汇编语言程序

在开始学习汇编语言程序设计之前,先给出一个完整的汇编语言程序,使读者对汇编语言程序有一个感性的认识。

【例 1.3】以下程序实现在显示器屏幕上输出一串字符“Hello!”。

程序如下:

```
data    segment                ;定义一个数据段
string  db 'Hello!$'          ;要输出的字符串 Hello!预先存放在以
                                ;string 开始的存储器单元中,以$结束
data    ends                  ;数据段定义结束
code    segment                ;定义一个代码段
main    proc far               ;定义一个主程序
        assume cs:code,ds:data ;对代码段和数据段寄存器进行说明
start:                                     ;主程序开始
;设置返回地址,
        push ds
        xor ax,ax
        push ax
;设置数据段寄存器
        mov ax,data
        mov ds,ax
;主程序部分
```

```

        lea dx,string           ;设置要输出的字符串的首地址
        mov ah,9               ;9号功能调用
        int 21h                ;使用9号系统功能调用输出一个字符串
        ret                    ;返回DOS操作系统
main    endp                  ;主程序结束
code    ends                  ;代码段结束
        end start              ;定义程序从start开始执行

```

在例 1.3 中, 汇编语言程序由两个段组成: 一是数据段 data, 二是代码段 code。数据段 data 中预存了要输出的字符串"Hello!", 代码段 code 则定义了将该字符串输出到显示器上的一段程序代码。该程序中包含了两类指令: 一是汇编伪指令, 二是汇编机器指令, 伪指令是由汇编程序解释的指令, 而机器指令才是真正由处理器执行的指令。程序的核心是一个名为 main 的主程序(又称为主过程), 它由从 push ds 指令开始到 ret 指令结束的 9 条汇编机器指令组成。

程序员首先使用编辑器将该段程序输入、编辑, 形成一个汇编语言源程序(扩展名为 .ASM), 然后使用一个汇编程序对该源程序进行汇编, 汇编程序对源程序进行以下处理:

- 对源程序进行语法、语义的检查, 若发现错误则指出错误所在。
- 对伪指令进行解释处理。
- 对汇编机器指令进行翻译, 转换为二进制机器指令。
- 生成一个可连接的二进制目标文件, 该二进制目标文件最终可通过连接程序的处理生成一个 .EXE 可执行文件。
- EXE 文件是可以被机器执行的程序文件, 只需在 DOS 状态下输入该文件名并按 Enter 键, 机器便开始执行该程序, 其正确结果是输出一串字符"Hello!"。

1.2 数据在计算机中的表示

早期的计算机主要用于科学计算, 以帮助人们提高计算的速度, 为此在计算机中必须使用一定的方法表示数值数据, 并完成各种运算。紧接着, 人们使用计算机来处理各种文字信息, 以提高对文本处理的效率, 为此在计算机中又必须使用一定的方法表示字符等非数值数据。随着计算机广泛应用于不同的领域, 计算机所能处理的信息种类越来越多, 如声音、图形、图像、视频和动画等。但无论什么样的信息或数据, 在计算机中都是通过二进制编码的方法来表示的。本节介绍数值数据、字符在计算机中的编码表示。

1.2.1 数值数据在计算机中的表示

数值数据是一种带符号数, 即有正负之分。在计算机中, 数的符号(+或-)和数的值一样都要采用二进制 0、1 编码。在计算机中通常采用补码编码来表示数值数据, 而数值数据又分为定点数和浮点数两种形式。

1. 数值数据的补码表示

补码表示法是计算机中表示带符号数常用的一种编码方法, 其符号位表示该数的正负号, 用“0”表示正(+)号, 用“1”表示负(-)号。

设纯小数的补码形式为 $x = x_0.x_1x_2 \cdots x_n$, 则补码表示的定义为:

$$[x]_{\text{补}} = \begin{cases} x & 0 \leq x < 1 \\ 2+x = 2 - |x| & -1 < x < 0 \end{cases}$$