

计算机科学与技术系列教材

汇编语言程序设计

主 编 何友鸣 苏光奎 石曙东 徐爱芸
副主编 王春梅 陈 宏 金大卫 徐 诚



WUHAN UNIVERSITY PRESS

武汉大学出版社

计算机科学与技术系列教材

汇编语言程序设计

主 编	何友鸣	苏光奎	石曙东	徐爱芸
副主编	王春梅	陈 宏	金大卫	徐 诚
参 编	王建红	鲍艳丽		

武汉大学出版社

图书在版编目(CIP)数据

汇编语言程序设计/何友鸣,苏光奎,石曙东,徐爱芸主编. —武汉:武汉大学出版社,2006.3

(计算机科学与技术系列教材)

ISBN 7-307-04941-4

I. 汇… I. ①何… ②苏… ③石… ④徐… III. 汇编语言—程序设计—教材 N. TP313

中国版本图书馆 CIP 数据核字(2006)第 005558 号

责任编辑:黄金文 夏炽元 责任校对:黄添生 版式设计:支笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北省孝感日报社印刷厂

开本:787×980 1/16 印张:14.875 字数:303千字

版次:2006年3月第1版 2006年3月第1次印刷

ISBN 7-307-04941-4/TP·193 定价:21.00元

版权所有,不得翻印;凡购我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

计算机科学与技术系列教材

编 委 会

主 任:何炎祥,武汉大学计算机学院院长,教授

副 主 任:康立山,中国地质大学(武汉)计算机学院院长,教授

陆际光,中南民族大学计算机科学学院院长,教授

编 委:(以姓氏笔画为序)

王江晴,中南民族大学计算机科学学院副院长,教授

王春枝,湖北工业大学计算机学院副院长,教授

牛冀平,黄冈师范学院计算机系主任,副教授

石曙东,湖北师范学院计算机科学与技术系主任,教授

朱 英,桂林电子工业学院计算机系副教授

孙扬波,湖北中医学院信息技术系信息管理与信息系统教研室
主任

刘腾红,中南财经政法大学信息学院副院长,教授

陈少平,中南民族大学电信学院副院长,教授

杜友福,长江大学计算机科学学院院长,教授

陆 迟,江汉大学数学与计算机科学学院计算机系主任,副教授

闵华松,武汉科技大学计算机科学与技术学院副院长,副教授

陈佛敏,咸宁学院信息工程学院计算机系主任,副教授

陈建新,孝感学院计算机科学系主任,副教授

李禹生,武汉工业学院计算机与信息工程系副主任,教授

李晓林,武汉工程大学计算机科学与工程学院副院长,副教授

张涣国,武汉大学计算机学院教授

余敦辉,湖北大学数学与计算机科学学院计算机系副主任

肖 微,湖北警官学院信息技术系副教授

钟 璐,武汉理工大学计算机科学与技术学院院长,教授

钟阿林,三峡大学电气信息学院计算机系主任

姜洪溪,襄樊学院电气信息工程系副主任,副教授

桂 超,湖北经济学院计算机与电子科学系副主任,副教授

黄求根,武汉科技学院计算机科学学院院长,教授

阎 菲,湖北汽车工业学院计算中心主任,副教授

韩元杰,桂林电子工业学院计算机系教授

谢坤武,湖北民族学院信息工程学院计算机系主任,副教授

戴光明,中国地质大学(武汉)计算机学院副院长,教授

魏中海,华中农业大学理学院计算机系副教授

执行编委:黄金文,武汉大学出版社副编审

内 容 简 介

汇编语言是汇编指令集、伪指令集和使用它们规则的统称。指令是组成本语言的主体;伪指令是在程序设计时所需要的一些辅助性说明指令。用汇编语言编写的程序称为汇编语言程序,或汇编语言源程序,在本教材中或特定的环境下,也可简称为源程序。汇编语言程序要比用机器指令编写的程序容易理解和维护,比用高级语言编写的程序效率高,它与硬件联系紧密。

本书知识层次清晰,系统地介绍了汇编语言的基本概念、基本原理和程序设计的基本方法。本书的宗旨是程序设计,因而附有大量的例题和习题,且都在机器上以NASM为支持系统运行通过。NASM是一个个人计算机上的超平台系统,即可以不选择操作系统而运行和执行汇编语言源程序。

为满足教学的实际需要,本书将编写配套的学习指导和实习指导书以及包括课件、例题、习题解答等内容的光盘。本书可作为大专院校计算机专业及相应专业本科、硕士研究生的计算机程序设计教材,也可以作相应层次的成人教育、职业教育的教材,亦可供计算机程序设计学习者、爱好者、职业人员或IT行业工程技术人员等学习参考。



前 言

汇编语言程序设计是计算机相关专业本、专科和研究生都必须学习和掌握的一门核心课程,也是学习操作系统、计算机接口技术和单片机技术的基础课程。

现代计算机的编程语言发展很快,高级语言的更新换代更是日新月异。尽管如此,许多计算机专业人士还是青睐于汇编语言。这是因为汇编语言是计算机系统提供给用户最快、最有效且能对硬件直接编程的语言,而且它的编程模型是 CPU 内部的寄存器、内存单元、CPU 的芯片端口等。因此,掌握了汇编语言,就了解了计算机的内部结构和 CPU 的工作流程。另外,在运用汇编语言进行程序设计的过程中,需要用到大量的计算机原理知识,这样就在学习和运用过程中使专业技能得到了相应的提高。

本书系统地介绍了汇编语言的基本概念、基本原理和程序设计的基本方法。另外本书还采用了实例教学的方法,通过大量实例讲述了如何用汇编语言开发应用程序,并介绍了上机调试运行汇编源程序的方法。这里以 NASM 为支持系统,NASM 是一个个人计算机上的超平台系统,即可以不选择操作系统而运行和执行汇编语言源程序。

本书分为七章和 2 个附录。第一章介绍汇编语言的一些基本概念,给出一些用汇编语言编程所需要的基本知识,包括数制的转换、计算机中编码及逻辑运算和信息在计算机中的表示方法。第二章介绍 NASM 汇编语言。本章介绍 NASM 汇编程序的开发过程以及 NASM 汇编程序中伪指令、有效地址、常量、表达式、临界表达式、局部标号、预处理器、目标文件格式等基本内容。第三章主要介绍 32 位处理器的指令系统。指令系统是一台计算机所有指令的集合,它确定了 CPU 所能完成的功能,本章的内容是学习本课程的重点和难点。通常,把指令按其功能分成以下几大类:数据传送指令、算术运算指令、BCD 码运算调整指令、逻辑运算指令、移位/循环指令、位处理指令、字符串操作指令、转移指令、处理器控制指令等。第四章介绍输入/输出设备。由于汇编语言具有直接控制硬件的特点,因而成为编写高性能输入/输出控制程序最有效的程序设计语言。第五章介绍在汇编语言程序设计中,最常见的程序设计方法:顺序程序设计、分支程序设计、循环程序设计以及子程序的设计等。这几种程序设计方法是汇编语言程序设计的基础。第六章主要讨论宏定义和宏调用的方法以及 C 语言和汇编语言的混合编程。第七章主要介绍文件处理的有关知识,以磁盘文件操作为主。本章介绍了一些有关文件 I/O 的基本概念,并讨论以文件代号方式和



BIOS INT 13H 所采用的磁盘文件访问方式。

本书附录 1 是 ASCII 码表,附录 2 介绍 NASM 汇编语言的汇编、连接操作方法以及建立 .exe 可执行文件的步骤,同时对汇编语言符号进行了说明,最后介绍了最常用的汇编指令集合。由于不同的 CPU 有不同的指令系统,所以这里不具体介绍某一种指令系统(需要时可参阅有关手册),只是概括和抽象出来,在 NASM 下可运行。

本书的知识层次清晰,各院校及其他教学单位完全可以视具体情况选讲。

本书第一章由湖北师范学院石曙东教授和王建红编写,第二章和附录由中南财经政法大学何友鸣教授、金大卫和鲍艳丽编写,第三章由武汉工程大学王春梅老师编写,第四章由江汉大学徐爱芸副教授编写,第五章由武汉大学苏光奎教授编写,第六章由武汉工程大学徐诚老师和中南财经政法大学的金大卫老师编写,第七章由桂林电子工业学院陈宏老师编写,最后由何友鸣教授总纂。我们衷心感谢中南财经政法大学信息学院院长、博士生导师杨云彦教授对本书的关心!衷心感谢兄弟院校的领导和学者和同仁们对本书的支持、肯定以及直接的帮助和中肯的、建设性的意见。

我们还要感谢武汉大学出版社的同志们,特别是黄金文、杨华二位副编审,是他们的艰辛努力和刻苦工作,才得以使本书早日与读者见面。

本书是我们长期教学工作的结果,所有的例题和习题都在机器上运行通过。尽管如此,肯定还存在着错误和不足,恳望诸位读者谅解和指正。如能赐教,不胜感谢!

最后,我们要由衷地感谢那些支持和帮助我们所有朋友们!谢谢你们使用和关心本书,并预祝你们教学或学习成功!

作 者

2006 年元旦于武昌

目 录

第一章 基础知识	1
1.1 汇编语言的由来和特点	1
1.1.1 机器语言	1
1.1.2 汇编语言	2
1.1.3 汇编程序和连接程序	2
1.1.4 汇编语言的主要特点	2
1.2 计算机中的数制	3
1.3 计算机中编码及逻辑运算	7
1.3.1 无符号数与带符号数	7
1.3.2 原码和补码	7
1.3.3 逻辑运算	9
1.3.4 BCD 码	10
1.3.5 ASCII 码	11
本章小结	12
习题一	12
第二章 NASM 相关知识	13
2.1 汇编程序的开发过程	13
2.1.1 NASM 汇编程序安装	13
2.1.2 NASM 汇编器	13
2.1.3 程序格式	14
2.1.4 一个显示字符串的汇编程序	14
2.2 伪指令	15
2.2.1 'DB' 一类的伪指令:声明已初始化的数据	15
2.2.2 'RESB' 类的伪指令:声明未初始化的数据	15
2.2.3 'INCBIN':包含其他二进制文件	16
2.2.4 'EQU':定义常数	16
2.2.5 'TIMES':重复指令或数据	16
2.3 有效地址	17



2.4 常量	17
2.4.1 数值常量	17
2.4.2 字符型常量	18
2.4.3 字符串常量	18
2.4.4 浮点常量	18
2.5 表达式	19
2.5.1 '!' :位或运算符	19
2.5.2 '^' :位异或运算符	19
2.5.3 '&' :位与运算符	19
2.5.4 '<<' and '>>' :位移运算符	20
2.5.5 '+' and '-' :加与减运算符	20
2.5.6 '*', '/', '//', '% '和'%% ' :乘除法运算符	20
2.5.7 一元运算符: '+', '-', '~'和'SEG'	20
2.6 临界表达式	20
2.7 局部标号	21
2.8 预处理器	22
2.8.1 最常用的方式: '% define'	22
2.8.2 取消宏定义: '% undef'	22
2.8.3 预处理器变量: '% assign'	23
2.8.4 多行宏: '% macro'	23
2.8.5 条件汇编	24
2.8.6 包含其他文件	24
2.8.7 汇编语言指引	24
2.9 目标文件格式	25
本章小结	25
习题二	25

第三章 数据操作	27
3.1 数据传送指令	27
3.1.1 通用数据传送指令	28
3.1.2 交换指令 (Exchange Instruction)	29
3.1.3 堆栈操作指令 (Stack Operation Instruction)	30
3.1.4 地址传送指令	31
3.1.5 I/O 指令	32
3.1.6 标志传送指令	33
3.2 算术运算指令	34

3.2.1 加法指令	34
3.2.2 减法指令	35
3.2.3 乘法指令	36
3.2.4 除法指令	37
3.2.5 类型转换指令	38
3.3 BCD 码运算调整指令	39
3.3.1 非压缩 BCD 码调整指令	39
3.3.2 压缩 BCD 码调整指令	41
3.4 逻辑运算指令	42
3.5 移位指令	44
3.5.1 算术移位指令	44
3.5.2 逻辑移位指令	44
3.5.3 双精度移位指令	45
3.5.4 循环移位指令	46
3.5.5 带进位的循环移位指令	46
3.6 位操作指令	47
3.6.1 位扫描指令	47
3.6.2 位测试和设置指令	48
3.7 标志控制指令	48
3.8 处理器指令	49
本章小结	49
习题三	49
 第四章 输入/输出程序设计	 52
4.1 工作原理	52
4.1.1 CPU 与外设的信息交换	52
4.1.2 CPU 寻址外设的方式	52
4.1.3 数据传送方式	53
4.2 数据的输入/输出方式	55
4.2.1 I/O 指令	55
4.2.2 程序查询 I/O 方式	56
4.2.3 中断 I/O 方式	57
4.3 基本输入	62
4.3.1 从键盘输入一个字符	63
4.3.2 从键盘输入一个字符串	64
4.4 基本输出	65



4.4.1 将一个字符输出到屏幕	65
4.4.2 将一个字符串输出到屏幕	65
4.4.3 将一个字输出到屏幕	66
4.5 键盘输入控制	69
4.6 屏幕输出控制	72
4.7 打印机输出控制	81
本章小结	86
习题四	87
第五章 程序设计的基本方法	89
5.1 程序设计的基本步骤	89
5.2 顺序程序设计	90
5.2.1 算术运算指令	90
5.2.2 逻辑运算指令	99
5.2.3 处理机控制指令	107
5.2.4 顺序程序设计举例	109
5.3 分支程序设计	109
5.3.1 控制转移指令	109
5.3.2 分支程序的结构形式	114
5.3.3 分支程序设计方法	115
5.4 循环程序设计	122
5.4.1 循环程序的结构	122
5.4.2 循环控制方法	123
5.4.3 单重循环程序设计	125
5.4.4 多重循环程序设计	127
5.5 子程序设计	127
5.5.1 子程序的概念	128
5.5.2 子程序的一般设计方法	129
5.5.3 嵌套与递归子程序	139
本章小结	142
习题五	143
第六章 高级编程	146
6.1 宏汇编	146
6.1.1 宏定义与宏调用	146
6.1.2 几个主要的宏操作伪指令	147



6.2 重复汇编和条件汇编	148
6.2.1 重复汇编	148
6.2.2 条件汇编	149
6.3 汇编语言和高级语言的混合编程	151
6.3.1 C语言调用汇编语言子程序	151
6.3.2 汇编语言程序调用C语言函数	153
6.4 汇编语言在 Visual C++ 中的应用	154
6.4.1 嵌入汇编语言指令	154
6.4.2 调用汇编语言过程	159
本章小结	164
习题六	164
第七章 文件处理	165
7.1 输入及输出概念	165
7.2 标准的文件代号	166
7.2.1 文件代号的建立	167
7.2.2 ASCII 串和文件属性设置	168
7.2.3 建立新文件	169
7.2.4 文件的打开和关闭	170
7.3 文件的读写操作	173
7.3.1 从文件或设备读取数据	174
7.3.2 数据写入文件或设备	177
7.3.3 移动文件指针	180
7.4 文件其他操作	187
7.4.1 检查并修改文件属性	187
7.4.2 删除一个文件	190
7.4.3 文件改名	191
7.5 目录操作	193
7.5.1 建立子目录	193
7.5.2 删除子目录	194
7.5.3 改变当前目录	194
7.5.4 取得当前目录	196
7.6 磁盘基本操作	196
7.6.1 取得缺省的磁盘驱动器	196
7.6.2 改变缺省的磁盘驱动器	197
7.7 面向 I/O 设备的文件操作	199



7.8 低级输入及输出	202
7.8.1 基本 BIOS INT 13H	203
7.8.2 扩展 INT 13H	207
本章小结	212
习题七	212
 附录 1 ASCII 码表	 214
附录 2 操作步骤和指令系统	215
 参考文献	 223



第一章 基础知识

【学习目的和要求】 本章要求大家掌握汇编语言的基本概念,并为下面的章节学习汇编语言做好准备。

1946年,在美国诞生了世界上第一台现代电子计算机 ENIAC,它标志着人类进入了计算机时代。从那时至今,计算机在人类的日常生活中扮演着越来越重要的角色,深入到了人类社会的各个角落。计算机是一种能够按照人们预先存放在其中的一系列命令连续高速地进行数据处理的电子机器。程序是用计算机语言编写的指令序列。能够把人的命令告诉计算机的一套符号系统及其使用规则称为“计算机语言”。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。其中,汇编语言是一种能够充分利用计算机硬件特性的低级语言,它与计算机的结构有非常紧密的联系。不同的计算机有各自不同的汇编语言,本书详细说明 Intel 系列 CPU 的汇编语言及其指令和编程。本章介绍汇编语言的一些基本概念和用汇编语言编程所需要的基础知识。

1.1 汇编语言的由来和特点

1.1.1 机器语言

机器指令是 CPU 能直接识别并执行的指令,它的表现形式是二进制编码。机器指令通常由操作码和操作数两部分组成,操作码指出该指令所要完成的操作,即指令的功能,操作数指出参与运算的对象,以及运算结果所存放的位置等。操作码在前,操作数在后。操作码部分用来指出本指令要求计算机做什么样的操作,是做加法,做减法,还是完成数据传送,抑或是其他的操作;操作数部分给出参与操作的数据值,或者指出操作对象在什么地方。由于机器指令与 CPU 紧密相关,所以,不同种类的 CPU 所对应的机器指令不同,而且它们的指令系统往往相差很大。但对同一系列的 CPU 来说,为了满足各型号之间具有良好的兼容性,一般要做到:新一代 CPU 的指令系统必须包括先前同系列 CPU 的指令系统。只有这样,先前开发出来的各类程序在新一代 CPU 上才能正常运行。机器语言是用来直接描述机器指令、使用机器指令的规则等,它是 CPU 能直接识别的惟一一种语言。也就是说,CPU 能直接执行用机器



语言描述的程序。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不易读、出错率高、难以维护,也不能直观地反映用计算机解决问题的基本思路。

1.1.2 汇编语言

虽然用机器语言编写程序有很高的要求和许多不便,但编写出来的程序执行效率高,CPU 严格按照程序员的要求去做,没有多余的额外操作。所以,在保留“程序执行效率高”的前提下,人们开始着手研究一种能大大改善程序可读性的编程方法。为了改善机器指令的可读性,选用了一些能反映机器指令功能的单词或词组的缩写来代表该机器指令,而不再关心机器指令的具体二进制编码。与此同时,也把 CPU 内部的各种资源符号化,使用该符号名也等于引用了该具体的物理资源。如此一来,令人难懂的二进制机器指令就可以用通俗易懂的、具有一定含义的符号指令来表示了,于是,汇编语言就有了雏型。现在,我们称这些具有一定含义的符号为助忆符,用指令助忆符、符号地址等组成的符号指令称为汇编格式指令(或汇编指令)。

汇编语言是汇编指令集、伪指令集和使用它们规则的统称。伪指令是在程序设计时所需要的一些辅助性说明指令,它不对应具体的机器指令,有关内容在以后的各章节中会有详细叙述,在此不展开介绍。用汇编语言编写的程序称为汇编语言程序,或汇编语言源程序,在本教材中或特定的环境下,也可简称为源程序。汇编语言程序要比用机器指令编写的程序容易理解和维护。

1.1.3 汇编程序和连接程序

汇编程序是一种计算机软件,属于系统软件部分,它能够把人们编写的汇编语言程序(称为源程序,一般以 ASM 作为文件扩展名)翻译成机器语言,这种翻译操作称为汇编。由于不同的计算机有不同的机器语言,因而也需要有不同的翻译器——汇编程序。NASM 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。另一方面,连接程序还具有把多个目标程序装配在一起的功能,也可以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。

1.1.4 汇编语言的主要特点

与机器语言相比,汇编语言易于理解和记忆,编写的源程序可读性较强。汇编语言提供了一些模块间相互连接的方法,因此,一个大的任务可以分解成若干模块,由



多个程序员分别设计,然后再汇总到一起,实现初步的结构化编程。源程序翻译成机器语言后的执行文件在存储空间、执行速度方面与机器语言编写的程序大致相当。高级语言在科学计算、事务处理等方面比汇编语言有巨大的优势,但用高级语言编写的程序,在翻译成机器语言后,程序代码冗长,占用存储空间大,执行速度慢。如果用高级语言来编写接口控制、设备通信等方面的程序则不太合适,相反,在这样的情况下汇编语言更容易发挥其长处:最终的执行代码简短,执行速度快,效率高,特别是汇编语言能直接控制计算机的外设,这些特点是高级语言和第四代语言望尘莫及的。

可见,高级语言适合于编写应用软件,而对于系统软件,尤其是涉及内存管理、硬件控制方面问题时汇编语言更合适。可以说,汇编语言程序设计是从事计算机研究与应用的重要手段,是软件与硬件相结合的基础。

1.2 计算机中的数制

1. 数制及其转换

人们在日常生活中每天都要与数打交道,我们通常书写的数是十进制数,而计算机内部使用的却是由 0 和 1 两种符号构成的二进制数。二进制数在书写时显得过于冗长、麻烦,所以在与计算机打交道时还会经常使用八进制数和十六进制数。同一个数值可以用不同的数制表示,不同数制之间可以相互转换。为了后面的需要,我们在此仅讨论整数在不同数制间的转换方法及其转换。

(1) 数制

任何一个数制都涉及到下面三个问题:

① 计数符号

这是用于书写数值的各个符号,所有计数符号构成的集合称作数符集。 k 进制的数符集中必然包含 k 个符号。比如:

二进制的数符集中有两个符号:0 和 1;

八进制的数符集中有 8 个符号:0,1,2,3,4,5,6,7;

十进制的数符集中有 10 个符号:0,1,2,3,4,5,6,7,8,9;

十六进制的数符集中有 16 个符号:0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。

显然,任意进制的各个计数符号是有顺序的。二进制、八进制和十进制的每个计数符号就是它的序号值,十六进制的后 6 个计数符号 A、B、C、D、E、F 的序号值依次是 10、11、12、13、14、15。

② 基数和权

如果把用 k 进制书写的一个整数的各个位从右往左依次记作第 0 位、第 1 位、…、第 n 位,则第 i 位上的数符 a_i 所代表的含义是 $a_i \times k_i$ 。在此, k 称为一个数制