

WUTP

面向21世纪
高职高专计算机类
专业新编系列教材

Assembly Language
Programming

汇编语言程序设计

主编 武新

武汉理工大学出版社

Wuhan University of Technology Press

WILEY

WILEY
SINCE 1807
PUBLISHED BY
JOHN WILEY & SONS

Assembly Language
Programming

汇编语言程序设计

王 颖 编著

清华大学出版社
Tsinghua University Press

面向 21 世纪高职高专计算机类专业新编系列教材

Assembly Language Programming

汇编语言程序设计

主 编	武 新	
副主编	余宏生	张永胜
	郑 敏	王永乐

武汉理工大学出版社

Wuhan University of Technology Press

内 容 提 要

本书是面向 21 世纪高职高专计算机类专业新编系列教材之一。该书主要阐述了 IBM-PC 及其兼容机宏汇编语言程序的设计和操作方法。全书共分 8 章,第 1 章介绍汇编语言基础知识;第 2 章介绍汇编语言的程序组织、常用伪指令、常用 DOS 调用及上机操作方法;第 3 章介绍各种寻址方式的应用、指令系统和指令的使用技巧;第 4 章介绍各种分支程序、循环程序的编程技巧、子程序的设计;第 5 章介绍字符串编程、宏的定义和使用方法、多模块程序设计方法、子程序库的建立和使用方法;第 6 章介绍中断程序的编写方法和使用技巧;第 7 章介绍 C 语言与汇编语言的混合编程方法;第 8 章介绍 80386/80486 和 Pentium 增加的指令。

本书可供高职高专计算机、自动化、机电类专业作为教材,也可供其他工程技术人员自学参考。

图书在版编目(CIP)数据

汇编语言程序设计/武新主编. —武汉:武汉理工大学出版社,2004. 8
面向 21 世纪高职高专计算机类专业新编系列教材
ISBN 7-5629-2119-9

I. 汇 II. 武 III. 汇编语言-程序设计 IV. TP313

中国版本图书馆 CIP 数据核字(2004)第 051485 号

出版发行:武汉理工大学出版社(武汉市洪山区珞狮路 122 号 邮政编码:430070)

HTTP://www.techbook.com.cn

E-mail:duanchao@mail.whut.edu.cn tiandq@mail.whut.edu.cn

经 销 者:各地新华书店

印 刷 者:武汉理工大印刷厂

开 本:787×960 1/16

印 张:16.25

字 数:319 千字

版 次:2004 年 8 月第 1 版

印 次:2004 年 8 月第 1 次印刷

印 数:1---5000 册

定 价:22.00 元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。本社购书热线电话:(027)87397097 87394412

凡使用本教材的教师,可拨打(027)87385610 免费索取电子教案光盘。

出版说明

面向新世纪,我国高等职业技术教育进入蓬勃发展的新时期。根据 IT 行业技术新、发展快的特点,高等专科学校、高等职业技术学院计算机类专业教育,按照社会主义市场经济规律的原则定位人才培养目标和调整教学方法,尽量按照新技术或新版本更新课程内容,加速各种新产品和新技术的推广应用,努力提升高等职业技术教育对国民经济发展的促进作用。

根据高等职业技术教育快速发展与教学改革对教材建设的需求,武汉理工大学出版社经过广泛调研,与国内近 30 所高等专科学校、高等职业技术学院的计算机教育专家进行探讨,决定组织编写一套适合于高等职业技术教育计算机类专业(涵盖计算机应用与维护、计算机网络技术、计算机软件技术等专业方向)人才培养和教学需要的具有特色的高质量教材——面向 21 世纪高职高专计算机类专业新编系列教材。

本套新编系列教材的编写具有以下特色:

1. 与时俱进,教材内容体现人才培养目标

本套教材的编写反映教育部制订的《高职高专教育基础课程教学基本要求》和《高职高专教育专业人才培养目标及规格》的文件精神,贯彻高等职业技术教育“要服务于社会主义现代化建设,要与生产劳动和社会实践相结合”的宗旨,以培养一大批满足生产第一线需要的高等技术应用型人才为目标,坚持以技术应用型为主线的原则来编写教材内容,加强应用能力的培养。

2. 紧跟教学改革步伐,体现教学改革阶段性成果

本套教材的编写反映高职高专学校教学改革的阶段性成果,在处理“基础理论”与“实践能力”之间的关系上,遵循“基础理论以够用、必需为度,突出应用”的原则。教材编写坚持“少而精”的原则,以培养从

事计算机应用与维护、网络建设与维护及软件开发与测试等方面的能力,并能够快速跟踪计算机新技术发展的高等技术应用型人才为目标。坚持理论与实际相结合,采用“提出问题—分析问题—设计任务—解决任务—总结规律”的编写方法,努力创造出高职高专教材新体系。

3. 实现立体化出版,适应教育方式的变革

本套教材努力使用和推广现代化的教学手段,凡有条件的课程都准备组织编写、制作和出版与教材配套使用的实验、习题、课件、电子教案及相应的程序设计素材库。

本套教材首批 26 种预计在 2004 年秋季至 2005 年春季全部出齐。我们的编审者、出版者决不敢稍有懈怠,一定高度重视,兢兢业业,按最高的质量标准工作。教材建设是我们共同的事业和追求,也是我们的共同的责任和义务,我们诚恳地希望大家积极选用本套教材,并在使用过程中给我们多提意见和建议,以便我们不断修订、完善全套教材。

武汉理工大学出版社

2004 年 1 月

面向 21 世纪高职高专计算机类专业 新编系列教材编审委员会

顾问:

钟 珞 危道军

主任委员:

舒云星 雷绍锋

副主任委员:(以姓氏笔画为序)

刘德清 李庆亮 张树臣 张浩军 周松林

郭长庚 徐卓峰 崔轩辉 常荆燕 黄春喜

委员:(以姓氏笔画为序)

丁文华 王一兵 王学军 王海芳 刘自强

孙清伟 宋锦河 李京秀 李晓桓 何月顺

陈 年 陈松才 陈桂生 陈 鑫 张有谊

张晓云 张新成 苏 玉 周 舸 金 平

武 新 欧晓鸥 赵丽梅 赵 静 姜华斌

徐立新 徐善荣 秦振吉 郭荣冰 黄亚平

崔晓军 戴春霞

秘书长:田道全

总责任编辑:段 超 徐秋林

前 言

汇编语言是目前能够最大限度发挥计算机硬件特性,并且满足苛刻的实时处理要求的最有效的语言。学习汇编语言离不开实际的计算机系统,当今广为流行的是以 Intel80X86 及 Pentium 为 CPU 的微型计算机系统,所以本书选用以 8086/8088 系列微机为主要背景机,以 8086/8088CPU 为基础的 IBM-PC 宏汇编语言 MASM 5. X。同时介绍 Intel80X86 及 Pentium 增加的指令功能。

本书本着由浅入深地组织教学内容的原则。对于基本知识和基本概念的介绍,结合应用性较强的实例由浅入深地介绍。另外还专门介绍一些汇编语言的应用程序,如奏乐程序、定位显示程序以及对文件的访问等程序以拓宽学生的思路。本书的另一个特点是各章节以例题开始,然后总结,书中习题全部上机调试通过。每章都有内容提要,最后有全章的小结。总之,本书的主要任务是使学生通过本课程的学习,了解计算机内部运行机制,掌握程序设计的方法和技巧。

全书共分八章,第 1 章介绍了汇编语言的基础知识,包括数据在计算机内的表示、计算机系统简介、Intel8086CPU 简介、Intel8086 的内存和堆栈、计算机语言概述。重点介绍了 CPU 中的寄存器及作用、数据在存储器中的存放形式;第 2 章介绍了汇编语言程序组织,首先通过两个完整的汇编语言源程序实例,介绍了汇编语言源程序的结构、语句格式及规则,还介绍了常用的伪指令、常用的 DOS 功能调用和上机操作全过程;第 3 章介绍了寻址方式与常用的汇编指令。重点在各种寻址方式的应用,指令的使用技巧及一些典型的实例;第 4 章介绍了基本结构程序设计方法,包括各种顺序程序设计、分支程序设计、循环程序的编程技巧、子程序的设计和使用方法;第 5 章介绍了高级汇编程序设计的方法。包括有关字符串操作的指令及程序设计、宏的定义、调用、扩展、宏库的使用、多模块程序设计的方法、子程序库的建立和使用方法;第 6 章介绍了中断程序设计及系统调用。包括设计用户自己的软中断程序、奏乐程序、定位显示程序以及对文件操作的程序实例;第 7 章介绍了汇编语言与 C 语言的连接,包括在 C 语言程序中直接嵌入汇编语言,C 语言程序直接调用汇编语言程序和汇编语言调用 C 函数的方法;第 8 章介绍了 80386、80486 和 Pentium 增加的汇编指令。

本书由武新担任主编并负责全书的统稿工作。其中第 1、2 章和附录由武新编写;第 3 章由郑敏编写;第 4 章 4. 1、4. 2 节由王淑敏编写,该章其余部分由王永乐编写;第 5 章由张永胜编写;第 6、7、8 章由余宏生编写。本书在编写前对大纲进行了充分的讨论,参与大纲讨论的有洛阳工业高等专科学校、黄石理工学

院、许昌职业技术学院、东华理工学院、四川理工学院、郑州高等专科学校、河南机电专科学校、襄樊职业技术学院、湖南商务职业技术学院、长江职业学院等。

虽然参加本书编写的都是多年从事汇编语言教学的人员,但由于作者水平有限,书中不妥或错误之处在所难免,恳切希望广大读者批评指正。

编 者

2004 年 5 月

目 录

1 汇编语言基础知识	(1)
1.1 数据在计算机内的表示	(1)
1.1.1 数值数据的表示	(1)
1.1.2 字符数据的表示	(3)
1.2 计算机系统简介	(5)
1.2.1 计算机的硬件	(5)
1.2.2 计算机的软件	(7)
1.3 Intel8086 CPU 简介	(7)
1.3.1 执行单元	(9)
1.3.2 总线接口单元	(11)
1.4 Intel8086 的内存和堆栈	(13)
1.4.1 内存存储器	(13)
1.4.2 堆栈	(13)
1.4.3 存储器物理地址的形成	(14)
1.5 计算机语言概述	(15)
1.5.1 机器语言	(16)
1.5.2 汇编语言	(16)
1.5.3 高级语言	(17)
1.5.4 各种语言的比较	(18)
习题与思考题	(19)
2 汇编语言程序组织	(20)
2.1 汇编源程序举例	(20)
2.1.1 源程序例题 1	(20)
2.1.2 源程序例题 2	(23)
2.2 汇编语言源程序结构	(27)
2.2.1 汇编语言的语句格式	(27)
2.2.2 汇编语言各语款项	(28)

2.3 常用伪指令	(30)
2.3.1 变量定义伪指令	(30)
2.3.2 赋值伪指令 EQU	(32)
2.3.3 等号伪指令 =	(33)
2.3.4 符号定义伪指令 LABEL	(33)
2.3.5 段定义伪指令 SEGMENT...ENDS	(33)
2.3.6 假定伪指令 ASSUME	(34)
2.3.7 置汇编地址计数器伪指令 ORG	(35)
2.3.8 结束伪指令 END	(35)
2.4 常用 DOS 输入输出功能调用	(36)
2.4.1 一个字符的输出	(36)
2.4.2 字符串的输出	(37)
2.4.3 键盘输入一个字符	(37)
2.4.4 控制台输入一个字符	(38)
2.4.5 键盘输入字符串	(38)
2.5 宏汇编语言程序的上机全过程	(39)
2.5.1 建立与修改汇编源程序(源程序文件的扩展名为 ASM)	(39)
2.5.2 对源程序文件汇编产生相应的目标文件(目标文件的扩展名为 OBJ)	(40)
2.5.3 连接目标文件生成可执行文件(可执行文件的扩展名为 EXE)	(43)
2.5.4 运行可执行文件	(45)
习题与思考题	(45)
实验题	(46)
 3 寻址方式与常用的汇编指令	 (47)
3.1 寻址方式	(47)
3.1.1 立即寻址(Immediate Addressing)	(47)
3.1.2 直接寻址(Direct Addressing)	(48)
3.1.3 寄存器寻址(Register Addressing)	(49)
3.1.4 寄存器间接寻址(Register Indirect Addressing)	(49)
3.1.5 变址寻址(Indexed Addressing)	(50)
3.1.6 基址加变址寻址(Based Indexed Addressing)	(51)
3.1.7 跨段问题	(52)
3.1.8 寻址方式综合例子	(53)
3.2 常用的指令系统	(54)
3.2.1 数据传送指令	(54)

3.2.2 算术运算类指令	(58)
3.2.3 逻辑运算与移位类指令	(63)
3.2.4 十进制算术运算指令	(68)
3.3 汇编语言中的表达式	(71)
3.3.1 数值表达式	(71)
3.3.2 地址表达式	(72)
3.3.3 运算符的优先级	(76)
习题与思考题	(78)
实验题	(82)
4 基本结构程序设计	(84)
4.1 顺序结构程序设计	(84)
4.1.1 顺序程序设计的结构形式	(85)
4.1.2 顺序程序举例	(86)
4.2 分支结构程序设计	(87)
4.2.1 转移指令	(89)
4.2.2 分支程序设计	(94)
4.2.3 树形分支程序设计	(97)
4.3 循环结构程序设计	(99)
4.3.1 循环程序的结构	(100)
4.3.2 循环程序的控制	(101)
4.3.3 多重循环程序的设计	(109)
4.4 子程序的设计	(111)
4.4.1 子程序的结构	(111)
4.4.2 子程序参数传递	(115)
4.4.3 子程序的嵌套	(120)
4.5 应用举例	(121)
习题与思考题	(126)
实验题	(128)
5 高级汇编程序设计	(129)
5.1 字符串操作	(129)
5.1.1 字符串传送指令	(130)
5.1.2 串比较指令	(133)
5.1.3 串搜索指令	(135)

5.1.4 串读取指令	(137)
5.1.5 串存储指令	(138)
5.2 宏结构程序设计	(141)
5.2.1 宏定义	(141)
5.2.2 宏调用与宏扩展	(142)
5.2.3 宏指令中参数的使用	(145)
5.2.4 宏库的建立与使用	(149)
5.2.5 宏指令与子程序的比较	(151)
5.3 多模块程序设计	(152)
5.3.1 多模块程序设计举例	(153)
5.3.2 模块之间的组合方式	(157)
5.3.3 模块之间的通信方式	(159)
5.4 子程序库的建立方法	(159)
5.5 汇编综合举例	(161)
习题与思考题	(168)
实验题	(168)
6 中断程序设计及系统调用	(170)
6.1 输入输出指令	(170)
6.1.1 I/O 端口	(170)
6.1.2 IN/OUT 指令	(170)
6.2 中断	(172)
6.2.1 中断概念	(172)
6.2.2 中断向量表	(175)
6.2.3 中断服务程序	(176)
6.3 常用 BIOS 与 DOS 中断调用	(179)
6.3.1 BIOS 与 DOS 中断调用概述	(179)
6.3.2 DOS 系统功能调用	(179)
6.3.3 BIOS 系统中断调用	(180)
6.4 磁盘文件管理	(183)
6.4.1 磁盘文件管理的基本要求	(183)
6.4.2 磁盘文件管理的功能调用	(184)
习题与思考题	(187)
实验题	(189)

7 汇编语言与 C 语言的连接	(190)
7.1 C 语言程序中直接嵌入汇编程序	(190)
7.1.1 嵌入 C 语言程序中的汇编语句的格式	(190)
7.1.2 汇编语句访问 C 语言的数据	(191)
7.1.3 嵌入汇编语句的 C 语言程序的编译过程	(192)
7.2 C 语言与汇编语言的混合编程	(194)
7.2.1 混合编程的约定规则	(194)
7.2.2 在 C 程序中直接调用汇编子程序	(196)
7.2.3 混合编程的参数传递	(198)
7.2.4 汇编语言程序对 C 语言程序的调用	(203)
习题与思考题	(206)
实验题	(206)
8 80X86 及 Pentium 汇编指令介绍	(208)
8.1 32 位微处理器	(208)
8.2 32 位扩展指令	(209)
8.2.1 数据传送类指令	(210)
8.2.2 算术运算类指令	(212)
8.2.3 位操作类指令	(214)
8.2.4 串操作类指令	(214)
8.2.5 控制转移类指令	(215)
8.2.6 高级语言支持	(215)
8.3 80386 新增指令	(216)
8.3.1 双精度移位指令	(216)
8.3.2 位扫描指令	(217)
8.3.3 位操作指令	(217)
8.3.4 条件设置指令	(218)
8.4 80486 新增指令	(219)
8.4.1 字节交换指令	(219)
8.4.2 交换加指令	(219)
8.4.3 比较交换指令	(219)
8.4.4 高速缓存(Cache)无效指令	(220)
8.4.5 回写及高速缓存无效指令	(220)
8.4.6 TLB 无效指令	(220)
8.5 Pentium 新增指令	(220)

8.5.1 8字节交换指令	(220)
8.5.2 处理器特征识别指令	(221)
8.5.3 读时间标记计数器指令	(222)
8.5.4 读模型专用寄存器指令 RDMSR	(222)
8.5.5 写模型专用寄存器指令 WRMSR	(223)
8.5.6 系统管理方式返回指令 RSM	(223)
习题与思考题	(223)
附 录	(225)
附录 A IBM-PC ASCII 码表	(225)
附录 B DEBUG 命令表	(226)
B.1 运行 DEBUG	(226)
B.2 DEBUG 命令的一些共同特点	(226)
B.3 DEBUG 命令	(226)
附录 C 8086 指令表	(229)
C.1 数据传送指令	(229)
C.2 算术运算指令	(230)
C.3 位操作指令	(232)
C.4 串操作指令	(233)
C.5 控制转移指令	(234)
C.6 处理机控制指令	(235)
附录 D 伪指令表	(236)
D.1 数据定义伪指令	(236)
D.2 符号定义伪指令	(237)
D.3 义段、组、过程的伪指令	(237)
D.4 程序模块的定义与通信伪指令	(238)
D.5 宏定义伪指令	(238)
D.6 其他伪指令	(239)
附录 E 汇编错误代码信息表	(239)
E.1 错误代码	(239)
E.2 未编号的错误信息	(244)
参考文献	(246)

1 汇编语言基础知识

本章提要

程序设计语言是开发软件的工具,它的发展经历了由低级到高级的过程。汇编语言是一种面向机器的低级语言。本章介绍了数据在计算机内的表示;计算机系统简介;Intel8086 CPU 简介;Intel8086 的内存和堆栈;计算机语言概述。重点要掌握 CPU 中的寄存器及作用;数据在存储器中的存放形式,特别要注意数值型数据高位数放高字节,低位数放低字节。难点是存储器物理地址的形成;数值数据的表示。

汇编语言以助记符的形式表示每一条指令,每条指令对应着计算机硬件的一个具体操作。利用汇编语言编写的程序与计算机硬件密切相关。程序员可以直接对处理器内部的寄存器、主存储器的存储单元以及外部设备接口电路的端口等进行操作,从而能够有效地控制硬件。汇编语言程序运行速度快、占用内存空间少,这些都是高级语言无法替代的。所以作为计算机专业人员,必须掌握汇编语言程序设计方法。

1.1 数据在计算机内的表示

1.1.1 数值数据的表示

在日常生活中一般采用十进制进行计数,但计算机中只能识别 0、1 代码,也就是说计算机中使用的是二进制计数。二进制数遵循“逢二进一”的原则,但是二进制数书写比较长,有时为了书写方便常用到八进制数和十六进制数,4 位二

进制数与八进制数、十六进制数、十进制数的对应关系如表 1.1 所示。

表 1.1 4 位二进制数与八、十六、十进制数的对应表

4 位二进制数	0000	0001	0010	0011	0100	0101	0110	0111
八进制数	00	01	02	03	04	05	06	07
十六进制数	0	1	2	3	4	5	6	7
十进制数	0	1	2	3	4	5	6	7

4 位二进制数	1000	1001	1010	1011	1100	1101	1110	1111
八进制数	10	11	12	13	14	15	16	17
十六进制数	8	9	A	B	C	D	E	F
十进制数	8	9	10	11	12	13	14	15

在汇编语言中,数字 12 用二进制表示为 1100B,八进制表示为 14Q(也可用 O 表示),十六进制表示为 CH。字母 B、Q、H 分别表示二进制数、八进制数和十六进制数。

1.1.1.1 有符号数的表示

数值数据可以用不同的码制来表示,常用的有原码、补码和反码表示法。由于做加减法运算时,补码表示法的符号位可以参加运算,并且不影响运算结果的正确性,所以计算机中的有符号数都采用补码表示法。使用最高有效位来表示数的符号,正数用 0 表示,负数用 1 表示。

(1) 正数的补码表示法

正数的补码表示为:最高位是 0,其它位就是该数的绝对值。

【例 1.1】 设 $M = +48 = +30H = 32 + 16 = 2^5 + 2^4$, 求 $[M]_{\text{补}}$ 。

则 M 的 8 位字长补码表示为: $[M]_{\text{补}} = 0011\ 0000B = 30H$

而 M 的 16 位字长补码表示为: $[M]_{\text{补}} = 0000\ 0000\ 0011\ 0000B = 0030H$

(2) 负数的补码表示法

负数的补码表示为:该数的正数补码表示按位取反后加 1。

【例 1.2】 设 $M = -48 = -30H$, 求 $[48]_{\text{补}} + [-48]_{\text{补}}$ 。

$[+48]_{\text{补}}$ 可表示为 $0011\ 0000B$

按位求反为 $1100\ 1111B$

加 1 后为 $1101\ 0000B$

则 M 的 8 位字长补码表示为: $[M]_{\text{补}} = 1101\ 0000B = 0D0H$

而 M 的 16 位字长补码表示为: $[M]_{\text{补}} = 1111\ 1111\ 1101\ 0000B = FFD0H$

8 位补码相加: $[48]_{\text{补}} + [-48]_{\text{补}} = [48 - 48]_{\text{补}} = [0]_{\text{补}} = 0011\ 0000B + 1101\ 0000B = 0000\ 0000B$

从上面的例子可以看出,补码表示正数时比较简单,就是该数的二进制表示形式。如 $[+1]_{\text{补}} = 0000\ 0001B$, $[+127]_{\text{补}} = 0111\ 1111B$, $[+0]_{\text{补}} = 0000\ 0000B$ 。

负数的补码表示相对麻烦一些,如 $[-1]_{\text{补}} = 1111\ 1111B$, $[-127]_{\text{补}} = 1000$