21st CENTURY
规划教材

面向21世纪高职高专计算机系列规划教材
COURSES FOR VOCATIONAL HIGHER EDUCATION: COMPUTER

汇编语言程序设计

ASSEMBLY LANGUAGE

贺亚茹 主 编

科学出版社
www.sciencep.com

面向21世纪高职高专计算机系列规划教材
COURSES FOR VOCATIONAL HIGHER EDUCATION: COMPUTER

汇编语言程序设计

贺亚茹 主 编
汪成龙 庞新法 张卫龙 副主编

科 学 出 版 社
北 京

内 容 简 介

本书立足于实用性、技能性，以 Intel 8086 CPU 指令系统与 Microsoft 宏汇编 6.0 为背景，简明扼要地介绍了汇编语言的基本理论和方法。全书共 10 章，分别介绍计算机语言基础知识、寻址方式与基本指令、汇编语言、分支程序设计、循环程序设计、子程序设计、汇编程序的输入/输出、宏汇编技术、I/O 程序设计、高级汇编程序设计等。全书提供了大量实例，几乎每章后都附有小结和习题。

在写作方法上，本书采用大量例题的形式，对汇编语言程序设计的基本方法和实际应用技术进行了透彻的讲解，突出技能性和应用性。

本书通俗易懂，可作为二年制或三年制高职高专计算机类各专业教材，也可供相关专业学生或夜大、电大、两大学生以及自学考试等人员参考使用。

图书在版编目(CIP)数据

汇编语言程序设计/贺亚茹主编. —北京：科学出版社，2005.3
(面向 21 世纪高职高专计算机系列规划教材)
ISBN 7-03-015054-6

Ⅰ.汇… Ⅱ.贺… Ⅲ.汇编语言-程序设计-高等学校：技术学校-教材 Ⅳ.TP313

中国版本图书馆 CIP 数据核字(2005)第 012854 号

责任编辑：万国清 孙露露 责任校对：马伟科
责任印制：吕春珉/封面设计：飞天创意

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com
双青印刷厂 印刷
科学出版社发行 各地新华书店经销
*
2005 年 3 月第 一 版 开本：787×1092 1/16
2005 年 3 月第一次印刷 印张：13 1/4
印数：1—3 000 字数：294 000

定价：18.00 元

(如有印装质量问题，我社负责调换<路通>)
销售部电话 010-62136131 编辑部电话 010-62138978-8004

面向21世纪高职高专规划教材专家委员会

主　任　李宗尧

副主任（按姓氏笔画排序）

丁桂芝　叶小明　张和平　林　鹏

黄　藤　谢培苏

委　员　（略）

信息技术系列教材编委会

主　任　丁桂芝

副主任（按姓氏笔画排序）

万金保　方风波　徐　红　鲍　泓

委　员（按姓氏笔画排序）

于晓平　马国光　仁英才　王东红　王正洪

王　玉　王兴宝　王金库　王海春　王爱梅

邓　凯　付百文　史宝会　本柏忠　田　原

申　勇　任益夫　刘成章　刘克敏　刘甫迎

刘经玮　刘海军　刘敏涵　安志远　许殿生

何瑞麟　余少华　吴春英　吴家碚　吴瑞萍

宋士银　宋锦河　张红斌　张环中　张海鹏

张蒲生　张德实　李云程　李文森　李　洛

李德家　杨永生　杨　闯　杨得新　肖石明

肖洪生　陈　愚　周子亮　周云静　胡秀琴

赵从军　赵长旭　赵动庆　郝　梅　唐铸文

徐洪祥　徐晓明　袁德明　郭庚麒　高延武

高爱国　康桂花　戚长政　曹文济　黄小鸥

彭丽英　董振珂　蒋金丹　韩银峰　魏雪英

出版前言

随着世界经济的发展，人们越来越深刻地认识到经济发展需要的人才是多元化、多层次的，既需要大批优秀的理论型、研究型的人才，也需要大批应用型人才。然而，我国传统的教育模式主要是培养理论型、研究型的人才。教育界在社会对应用型人才需求的推动下，专门研究了国外应用型人才教育的成功经验，结合国情大力度地改革我国的“高等职业教育”，制定了一系列的方针政策。联合国教科文组织1997年公布的教育分类中将这种教育称之为“高等技术与职业教育”，也就是我们通常所说的“高职高专”教育。

我国经济建设需要大批应用型人才，呼唤高职高专教育的崛起和成熟，寄希望于高职高专教育尽快向国家输送高质量的紧缺人才。近几年，高职高专教育发展迅速。目前，各类高职高专学校已占全国高等院校的近1/2，约有600所之多。教育部针对高职高专教育出台的一系列政策和改革方案主要体现在以下几个方面：

- “就业导向”成为高职高专教育的共识。高职高专院校在办学过程中充分考虑市场需求，用“就业导向”的思想制定招生和培养计划。
- 加快“双师型”教师队伍建设。已建立12个国家高职高专学生和教师的实训基地。
- 对学生实行“双认证”教育。学历文凭和职业资格“双认证”教育是高职高专教育特色之一。
- 高职高专教育以两年学制为主。从学制入手，加快高职高专教学方向的改革，充分办出高职高专教育特色，尽快完成紧缺人才的培养。
- 开展精品专业和精品教材建设。已建立科学的高职高专教育评估体系和评估专家队伍，指导、敦促不同层次、不同类型的学校办出一流的教育。

在教育部关于“高职高专”教育思想和方针指导下，科学出版社积极参与到高职高专教材的建设中去，在组织教材过程中采取了“请进来，走出去”的工作方法，即由教育界的专家、领导和一线的教师，以及企事业从事人力资源工作的人员组成顾问班子，充分分析我国各地区的经济发展、产业结构以及人才需求现状，研究培养国家紧缺人才的关键要素，寻求切实可行的教学方法、手段和途径。

通过研讨认识到，我国幅员辽阔，各地区的产业结构有明显的差异，经济发展也不平衡，各地区对人才的实际需求也有所不同。相应地，对相同专业和相近专业，不同地区的教学单位在培养目标和培养内容上也各有自己的定位。鉴于此，适应教育现状的教材建设应该具有多层次的设计。

为了使教材的编写能针对受教育者的培养目标，出版社的编辑分不同地区逐所学校拜访校长、系主任和老师，深入到高职高专学校及相关企事业，广泛、深入地和教学第

一线的老师、用人单位交流，掌握了不同地区、不同类型的高职高专院校的教师、学生和教学设施情况，清楚了各学校所设专业的培养目标和办学特点，明确了用人单位的需求条件。各区域编辑对采集的数据进行统计分析，在相互交流的基础上找出各地区、各学校之间的共性和个性，有的放矢地制定选题项目，并进一步向老师、教育管理者征询意见，在获得明确指导性意见后完成“高职高专规划教材”策划及教材的组织工作：

- 第一批“高职高专规划教材”包括三个学科大系：经济管理、信息技术、建筑。
- 第一批“高职高专规划教材”在注意学科建设完整性的同时，十分关注具有区域人才培养特色的教材。
- 第一批“高职高专规划教材”组织过程正值高职高专学制从3年制向2年制转轨，教材编写将其作为考虑因素，要求提示不同学制的讲授内容。
- 第一批“高职高专规划教材”编写强调
  - 以就业岗位对知识和技能需求下的教材体系的系统性、科学性和实用性。
  - 教材以实例为先，应用为目的，围绕应用讲理论，取舍适度，不追求理论的完整性。
  - 提出问题→解决问题→归纳问题的教、学法，培养学生触类旁通的实际工作能力。
  - 课后作业和练习（或实训）真正具有培养学生实践能力的作用。

在“高职高专规划教材”编委的总体指导下，第一批各科教材基本是由系主任或从教学一线中遴选的骨干教师执笔撰写。在每本书主编的严格审读及监控下，在各位老师的辛勤编撰下，这套凝聚了所有作者及参与研讨的老师们的经验、智慧和资源，涉及三个大的学科近200种的高职高专教材即将面世。我们希望经过近一年的努力，奉献给读者的这套书是他们渴望已久的适用教材。同时，我们也清醒地认识到，“高职高专”是正在探索中的教育，加之我们的水平和经验有限，教材的选题和编辑出版会存在一些不尽人意的地方，真诚地希望得到老师和学生的批评、建议，以利今后改进，为繁荣我国的高职高专教育不懈努力。

科学出版社

2004年6月1日

前　言

“汇编语言程序设计”是二年制和三年制高职高专计算机应用、软件开发技术等计算机类各专业的一门专业基础课程。在编写过程中，本着突出实用性、技能性的原则，考虑到高职高专学生的文化基础知识结构特点和就业需要，以及学制短、教学时数有限的实际情况，并结合编者从事高职高专教学的实际经验，对全书所讲内容进行了重新编排，以便二年制和三年制高职高专院校根据不同的教学时数选讲其中的有关章节。

本书的主要内容和特点可归纳如下：

1）对于二年制和三年制高职高专学生都必须掌握的汇编语言基础知识部分（第1～6章），采用大量例题的形式，先做详细分析再进行讲解，并配以相应的习题，使读者能尽快掌握汇编语言程序设计的基本方法。

2）第7～10章为汇编语言实际应用部分，其中第7章介绍键盘输入、屏幕输出的汇编语言程序设计方法；第8章介绍宏汇编技术，为后面的学习奠定基础；第9章介绍设备接口程序的汇编语言程序设计方法；第10章通过对汇编语言与C/C++语言的比较，介绍汇编语言与C/C++之间混合编程技术。这部分内容力求突出实用性，以便满足学生实际工作的需要。

3）教材内容便于二年制和三年制高职高专院校分别选讲，如第8章宏汇编技术、第10章高级汇编程序设计部分和各章的一些例题等，均可根据各院校的实际情况选择讲授。

本书中的所有例题均经过上机调试。

本书通俗易懂，可作为二年制或三年制高职高专计算机类各专业教材，也可供相关专业学生或夜大、电大、函大学生以及自学考试等人员参考使用。

本书由贺亚茹担任主编，由汪成龙、庞新法、张卫龙担任副主编，姚卫国、车飞锋对本书中的所有例题和习题进行了认真调试，廖娜、薛慧芳、黄临娜、王立红参加了本书的讨论和文字录入工作。

由于编者水平有限，书中难免存在不足和疏漏之处，希望广大读者批评指正。

目　录

第 1 章　概　　述

知识点

- 计算机语言分类及特征
- 计算机中数据的表示
- CPU 内部结构
- 内存的组织

难点

- 数的补码表示
- CPU 寄存器组
- 内存分段及地址转换

要求

掌握

- 数的补码表示
- 数值转换
- CPU 寄存器组
- 逻辑地址到物理地址转换

了解

- 语言分类及特征
- CPU 内部结构

1.1　计算机语言分类及特征分析

人与人之间交流思想，必须通过二者都能够相互理解的语言来完成。同样，人和计算机之间交换信息也是如此。也就是说，人与计算机之间要实现信息交换，必须通过人机都能够相互理解的语言来完成。把计算机能够理解并执行的语言统称为计算机语言。到目前为止，计算机语言已经历了 4 代，分别是机器语言、汇编语言、高级语言、第 4 代语言。其中汇编语言是一种能够充分利用计算机硬件特征的低级语言，它与计算机的硬件密切相关。不同的计算机有各自的汇编语言，本书介绍 Intel 8086/8088 的汇编语言。

1.1.1　机器语言

计算机的原始定义是用电子线路来实现数学运算的机器，计算机内部全是电子线路，一条线上要么有电，要么无电，只有两个状态，有电即为 1，无电即为 0。计算机

发展到今天，已经具有计算、文字处理、游戏、播放 VCD 等功能。

机器语言：把由 0、1 代码构成的计算机能够认识且执行的语言叫做机器语言。任何语言编写的程序最终都必须翻译成机器语言。同自然语言一样，任何语言从语法上讲，都必须有主语、谓语、宾语、状语，其中谓语表示做什么，宾语表示对谁做，状语表示怎么做等。机器语言同样如此，设计一个机器的机器语言时，必须体现出做什么，对谁做，至于主语则是用户，可省掉不要，故机器语言语法格式为

做什么	对谁做

其中“做什么”回答了做何操作，且必须用 0、1 代码表示，故在机器语言中，把用 0、1 代码表示的做何操作的代码叫做操作码。“对谁做”回答了对哪些数做此操作，把参加操作的数叫操作数，这样一来，机器语言的格式演变为

操作码	操作数

下面以假想机为例，说明机器语言的格式，假设操作码占 4 位，两个操作数各占 3 位，故一条指令只占 10 位。用 0000 表示加法，用 0001 表示减法，用 0010 表示乘法等，故该 CPU 所能认识的指令最多 $2^4=16$ 条，则指令 0000 000 010 表示 000 单元的数和 010 单元的数作加法运算，运算的结果送回 000 单元，回答了怎么做（运算后的结果怎么办）。这样，完整的指令格式为

操作码	目的操作数	源操作数

其中，目的操作数既是参加运算的数，又是存放结果的数，源操作数仅是参加运算的数。上述是一个假想机的机器指令，用 0000 代表加法，0001 代表减法等。在设计 CPU 时，不同的线路表示不同的设计，对于别的 CPU，0000 可能表示减法，0001 可能表示加法。所以说，机器语言是面向机器的语言，是一台计算机的语言，更严格地讲，是一个 CPU 的语言，故本教材只讲 8086/8088 的汇编语言。PC 机系列是兼容机，假定 8086 CPU 有 100 条指令，在开发 80286 时，先保证 80286 能“理解”此 100 条指令，在此基础上，再增加若干条指令即为 80286，因此，在 8086 上能够运行的程序，一定可以在 80286 上运行，反之则不然。

依上所述，机器语言有以下特征：

1）是唯一的能被计算机识别并执行的语言。

2）是由 0、1 代码构成的语言，和自然语言相差甚远，不便于阅读和理解。

3）是面向机器的语言（低级语言）。

1.1.2 汇编语言

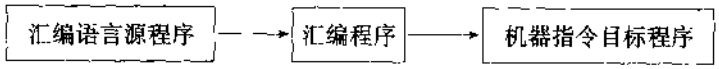
从 1.1.1 节可以看出，机器语言是由 0、1 代码构成，与自然语言相差甚远，即使是计算机专业人员也很难看懂，很难记忆，那么能否采用一套有助于用户记忆的符号（助记符）来表示此语言呢？这就是汇编语言。

汇编语言：0、1 代码机器语言的助记符表示叫做汇编语言。例如，机器指令 0000 101 011 很难记忆，其助记符为 ADD 5，3。其中 ADD、5、3 都很容易记忆。这样一

来，用户编程、记忆很容易，较接近于自然语言。但是用汇编语言书写的程序，CPU又不认识，解决办法是将汇编语言语句通过软件（汇编程序）翻译（转换）成机器语言。

汇编程序：把汇编语言书写的源程序翻译成机器指令所表示的目标程序，该翻译程序就叫汇编程序，反之，即为反汇编程序。

汇编语言源程序：用汇编语言书写的程序称为汇编语言源程序，简称源程序，它是汇编程序加工的原始数据，汇编程序加工的结果叫目标程序，该目标程序是由机器指令构成，故又称为机器指令目标程序，三者的关系为



依上所述，汇编语言有以下特征：

1）机器指令的助记符表示，较接近自然语言，容易编程、阅读和记忆。

2）翻译程序是一对一的转换，生成的目标代码效率高（时空性能好）。

3）适合于在硬件层次上开发程序。

1.1.3 高级语言

高级语言是一种脱离计算机，面向过程，更符合人们的思维，易为人们所学习和理解的语言。高级语言程序由语句组成，每条语句的功能相当于若干条机器指令，也就是说，一条高级语言语句相当于几十条机器指令，因此高级语言语句功能更强，编程更容易。目前常用的高级语言有BASIC、FORTRAN、PASCAL、C语言等。

同样道理，用高级语言书写的源程序也必须翻译成机器指令目标程序。完成此翻译任务的程序称为编译程序。这样一来，编译程序和汇编程序好像差别不大，但汇编程序是一对一的转换，而编译程序则是一对多的转换。例如：

$$a=2+3+9+8 \xrightarrow{\text{等价}} \begin{array}{l} \text{MOV} \quad A,0 \\ \text{ADD} \quad A,2 \\ \text{ADD} \quad A,3 \\ \text{ADD} \quad A,9 \\ \text{ADD} \quad A,8 \end{array}$$

依上所述，高级语言有以下特征：

1）更接近于自然语言，编程、阅读更容易。

2）与计算机硬件无关，一个机器是否支持该高级语言，只取决于有无相应的编译软件。

3）生成的目标代码效率低（时空性能差）。

1.2 数据类型及其内部表示

1.2.1 数制及其转换

数制也称为进位计数制，在日常生活中，人们常用的数制有以下几种：

1）二进制：由数字0、1构成，逢2进1。

2）八进制：由数字 0～7 构成，逢 8 进 1。

3）十进制：由数字 0～9 构成，逢 10 进 1。

4）十六进制：由数字 0～9 和字母 A～F 构成，逢 16 进 1。

5）六十进制：逢 60 进 1，如 1 小时＝60 分，1 分＝60 秒。

计算机仅能识别 0、1 代码，计算机能够理解的语言只能是由 0、1 构成的语言。计算机处理的数据（数值数据、字符、图形、声音等）必须用 0、1 代码表示，即二进制数据表示。对于一个数据，用户在书写时可以用任何进制表示，但在计算机内部处理时，都必须转换（翻译成）二进制数。

1．十进制数转换为二进制数

对于十进制数 8888，虽然每位均是 8，但 8 所处的位置不同，其权值不同。因此，我们可以把该数表示成如下形式：

$$8888=8\times10^3+8\times10^2+8\times10^1+8\times10^0$$

下面给出十进制数转换为二进制数的算法。

假定$(a_na_{n-1}\cdots a_0)_{10}=(b_mb_{m-1}\cdots b_0)_2$，则按权展开为

$$(a_na_{n-1}\cdots a_0)_{10}=b_m\times2^m+b_{m-1}\times2^{m-1}+\cdots+b_0$$

其中，b_0等于 $a_na_{n-1}\cdots a_0$的余数，其商为 $b_m\times2^{m-1}+b_{m-1}\times2^{m-2}+\cdots+b_1=(a_na_{n-1}\cdots a_0/2)$。$b_1$等于$(a_na_{n-1}\cdots a_0/2)/2$的余数，依此类推。

算法小结：用 2 除，直至商为 0，其余数的倒序即为相应的二进制数。

【例 1.1】 将十进制数 25 转换成二进制数。

解：

2	25	余数
2	12	……1
2	6	……0
2	3	……0
2	1	……1
	0	……1

$(25)_{10}=(11001)_2$

把十进制纯小数转换成二进制纯小数。假定：

$$(0.a_na_{n-1}\cdots a_0)_{10}=(0.b_mb_{m-1}\cdots b_0)_2$$

按权展开为

$$0.a_na_{n-1}\cdots a_0=b_m\times2^{-1}+b_{m-1}\times2^{-2}+\cdots b_1\times2^{-m}+b_0\times2^{-(m+1)}$$

其中，b_m等于$0.a_na_{n-1}\cdots a_0\times2$的整数部分，小数部分为 $b_{m-1}\times2^{-1}+b_{m-2}\times2^{-2}+\cdots b_1\times2^{-m+1}+b_0\times2^{-m}$。$b_{m-1}$等于$(0.a_na_{n-1}\cdots a_0\times2)\times2$，依此类推。

算法小结：用 2 乘，直至小数部分为 0，其乘积的整数部分顺序排列，即为相应二进制小数的小数部分。

【例 1.2】 将十进制纯小数 0.3125 转换成二进制数。

解：

0.3125		整数部分
× 2		
0.625	……0	
× 2		
0.25	……1	
× 2		
0.5	……0	
× 2		
0.0	……1	

故$(0.3125)_{10}=(0.0101)_2$

2. 二进制数转换为十进制数

将二进制数转换为十进制数，只需按权展开即可。

【例 1.3】 把二进制数 11001.0101 转换为十进制数。

解：

$$\begin{aligned}11001.0101 &= 1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0\\&\quad+0\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}\\&=16+8+0+0+1+0+0.25+0+0.0625\\&=(25.3125)_{10}\end{aligned}$$

任意进制数与十进制数转换的一般方法如图 1.1 所示。

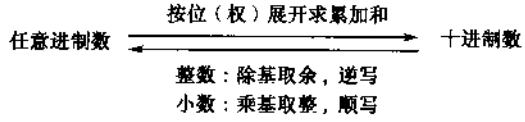


图 1.1 任意进制数与十进制数之间转换关系

3. 二进制数与十六进制数的相互转换

二进制数转换成十六进制数比较容易，具体方法如下：

1）把二进制数自右向左每 4 位分成一组，最左边不足 4 位的左补 0。

2）把每组 4 位的二进制数转换成 1 位的十六进制数。

3）按从左到右的次序写出转换结果。

【例 1.4】 把二进制数 101100110101111 转换成十六进制数。

解： 分组：左补一个 0：0101，1001，1010，1111

转换： 5 9 A F

所以，等值的十六进制数是 59AF。

十六进制数转换成二进制数的方法更简单，只需从左到右把每位十六进制数写成相应的 4 位二进制数，不足 4 位则左补 0 凑齐 4 位，并把结果写在一起即可。

【例 1.5】 把十六进制数 3BD 转换成二进制数。

解： 等值的二进制数是 001110111101，去掉最左边没有意义的 0，得 1110111101。

表 1.1 中列出了 0～15 之间的十进制数在二进制、八进制和十六进制下的对应值。为了加快数制转换的速度，这张表中的内容应该熟记于心。二进制、十六进制下多位数的加减法也应作为基本技能熟练掌握。

表 1.1 0～15 在各数制下的表示

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	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

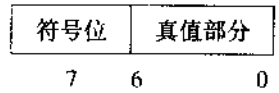
4. 数的书写方法

人们喜欢用十进制，而计算机仅认识二进制。十六进制、八进制是二进制的缩写，看到了十六进制、八进制，就等于看到了二进制。故计算机中经常使用的进制有二进制（后缀 B，Binary）、八进制（后缀 O 或 Q，octal，因 O 与 0 容易混淆，所以一般用 Q）、十进制（后缀 D，Decimal，或者不要后缀）和十六进制（后缀 H，Hex）。

例：
```
1010B      ；二进制
1010Q      ；八进制
1010H      ；十六进制
1010D      ；十进制
1010       ；十进制
```

1.2.2 符号数的表示方法

一个数前面冠以“+”或“-”的符号，表示该数为正数或负数，这样的数称为符号数。例如数 +125、-123 等。从此两例可以看出，数由两部分构成，符号位和真值部分，且在计算机内部必须用 0、1 代码表示。符号位要么是“+”，要么是“-”，只有 2 个状态，故用一个二进制位表示，0 表示“+”，1 表示“-”，其余部分为真值部分。因此符号数的存储格式为（机器字长假定为 8 位）



对于符号数，在计算机中通常采用两种办法表示（存储），原码和补码。

1. 原码

用原码表示符号数，最高位表示符号位，其余位是真值的原二进制代码，故称原码