HULUNBEIER XUEYUAN SHIERWU GUIHUA JIAOCAI CONGSHU
呼伦贝尔学院十二五规划教材丛书

HUIBIAN YUYAN CHENGXU SHEJI

汇编语言程序设计

主编 木林

东北师范大学出版社
Northeast Normal University Press

HULUNBEIER XUEYUAN SHIERWU GUIHUA JIAOCAI CONGSHU
呼伦贝尔学院十二五规划教材丛书

HUIBIAN YUYAN CHENGXU SHEJI

汇编语言程序设计

主 编 木 林

编 者 [illegible] 威

[illegible] 阿拉坦其其格

东北师范大学出版社

长春

图书在版编目(CIP)数据

汇编语言程序设计/木林主编. —长春:东北师范大学出版社,2012.11
ISBN 978-7-5602-8667-9

Ⅰ.①汇… Ⅱ.①木… Ⅲ.①汇编语言—程序设计
Ⅳ.①TP313

中国版本图书馆 CIP 数据核字(2012)第 284544 号

□策划编辑:魏 昆
□责任编辑:魏 昆 □封面设计:曹 涛 刘 萌
□责任校对:曲 颖 □责任印制:张允豪

东北师范大学出版社出版发行
长春净月经济开发区金宝街 118 号(邮政编码:130117)
电话:0431—85687213
传真:0431—85691969
网址:http://www.nenup.com
电子函件:sdcbs@mail.jl.cn
东北师范大学出版社激光照排中心制版
长春市东方票证印务有限责任公司印制
长春市高新区创举街705号(130012)
2012 年 11 月第 1 版 2012 年 11 月第 1 次印刷
幅面尺寸:185 mm×260 mm 印张:21.25 字数:485 千

定价:45.00 元

内 容 简 介

本书以 80×86CPU 为基础，由浅入深地介绍了汇编语言程序相关知识。全书共 12 章，主要介绍计算机组织结构和基本知识，PC 机指令系统和寻址方式，汇编语言程序格式，程序结构，I/O 程序设计，BIOS 和 DOS 中断调用，外部设备访问，模块化程序设计，汇编语言和高级语言混合编程。通过这些内容的学习，可掌握汇编语言编程基本原理、方法与技术，具备汇编语言软件开发应用的能力。

本书注重理论与实践的结合，内容新颖，材料丰富，对开发实例提供完整的程序代码。

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。

前　言

汇编语言程序设计是计算机专业的专业必修课中的核心课程。它不仅是计算机组成原理、操作系统、计算机接口电路、单片机等核心课程的必要先修课，而且对于训练学生掌握程序设计、熟悉上机操作和程序调试技术有着重要作用。

本书以80×86 CPU为基础，由浅入深地介绍了汇编语言程序相关知识。全书共12章。第1章，预备知识，主要介绍计算机组织结构和基本知识；第2章，CPU资源和存储器；第3章，操作数的寻址方式；第4章，微机指令系统；第5章，常用伪指令和操作符；第6章，基本程序设计，包括顺序程序设计、分支程序设计、循环程序设计和子程序设计；第7章，串操作指令与程序设计；第8章，输入输出和中断，BIOS和DOS中断调用，外部设备访问；第9章，宏；第10章，应用程序的设计；第11章，汇编语言和高级语言混合编程；第12章，汇编语言调试工具。通过这些内容的学习，掌握汇编语言编程基本原理、方法与技术，具备汇编语言软件开发应用的能力。

汇编语言作为一门重要的计算机课程，教学活动由理论教学与实验教学两部分构成。其最显著的特点是与机器硬件紧密相关，汇编指令与机器指令一一对应，是一门软件和硬件紧密结合的课程。但现有汇编语言程序设计方面的教材，大多注重汇编指令的介绍，对实验环节涉及较少，并且不能将理论与实际操作有机结合起来。因此本教材通过实验课程在整个教学活动中的体现，让学生通过实验深入了解CPU、存储器、接口及外部设备的工作特性，对数据在计算机中的表示和传送产生感性认识。实验活动将算法的实现和计算机的操作过程相结合，促成算法的逻辑意义与机器的实际操作的完整统一，使学生体会到算法的逻辑演义实际上是通过计算机的实际过程实现的。

我们根据汇编语言程序设计课程教学大纲的要求，在编写过程中，认真研究本科汇编语言课程教学的特点，并对国内外教材进行了反复比较，努力做到编写指导思想和编写思路明确清晰，教材特色明显，编写主要侧重理论与实践的结合。其中，理论部分主要介绍汇编语言的指令、原理、规则、程序，分析汇编语言的结构、编程方法；实验部分在每章都有体现，涉及实验内容、实验环境、实验调试工具、实验程序代码的介绍，这样把理论与实践有机地结合在一起。

本书编者多年从事汇编语言教学与科研工作，对汇编语言的教学与应用有深刻的理解和丰富的经验。本书由木林担任主编，于世华、庄新妍、李威、阿拉坦其其格参编，最后木林统一定稿。全书由庄新妍编写第一、二、四章，李威编写第三、五章，木林编写第六、七、八章，阿拉坦其其格编写第十章，于世华编写第九、十一章、十二章及附录。

在编写的过程中，编者力求精益求精，但由于编者水平有限，书中难免有错误和不妥之处，恳请广大读者批评指正。

编者

2012.6.19

【目　录】

【第1章】

预 备 知 识

本章介绍汇编语言的一些基本概念，给出汇编语言编程所需要的基本知识。

1.1 汇编语言的由来及其特点

1.1.1 机器语言

说到汇编语言的产生，首先要讲一下机器语言。机器语言是机器指令的集合。机器指令展开来讲就是一台机器可以正确执行的命令。电子计算机的机器指令是一列二进制数字。计算机将之转变为一列高低电平，以使计算机的电子器件受到驱动，进行运算。

上面所说的计算机指的是可以执行机器指令，进行运算的机器。这是早期计算机的概念。现在，在常用的PC机中，有一个芯片来完成上面所说的计算机的功能。这个芯片就是我们常说的CPU（Central Processing Unit，中央处理单元）。CPU是一种微处理器。以后我们提到的计算机是指由CPU和其他受CPU直接或间接控制的芯片、器件、设备组成的计算机系统。

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。

由于机器指令与CPU紧密相关，所以不同的CPU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。所以每一种微处理器都有自己的机器指令集，也就是机器语言。

早期的程序设计均使用机器语言。用机器语言编写程序的人一般都是经过严格训练的专业技术人员，普通的程序员一般都难以胜任。而且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。

要在显示器上输出“hello，world.”机器码如下

0B5BH：0000H 0110 1000 0110 0101 0110 1100 0110 1100 0110 1111 0010 1100
0111 0111 0110 1111 0111 0010 0110 1100 0110 0100 0010 1110
0000 1101 0000 1010 0010 0100

0B5CH：0000H1011 1000 0101 1011 0000 1011

0B5CH：0003H1000 1110 1101 1000

0B5CH：0005H1000 1101 0001 0110 0000 0000 0000 0000

0B5CH：0009H1011 0100 0000 1001

0B5CH：000BH1100 1101 0010 0001

0B5CH：000DH1011 1000 0000 0000 0100 1100

0B5CH：0010H1100 1101 0010 0001

看到这样的程序，读者会有什么感想？如果程序里有一个“1”被误写为“0”，又如何去查找呢？

1.1.2 汇编语言

为了改善机器语言的可读性，人们选用了一些能反映机器指令功能的单词或词组来代表该机器指令，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。在编写程序时，使用该符号名相当于引用了该具体的物理资源。于是产生了汇编语言。

汇编语言的主体是汇编指令。汇编指令和机器指令的差别在于指令的表示方法上。汇编指令是机器指令便于记忆的书写格式。

例如：机器指令 1000 1001 1101 1000 表示把寄存器 BX 的内容送到寄存器 AX 中。汇编指令则写成 mov ax，bx。这样的写法与人类语言接近，便于阅读和记忆。

（寄存器：CPU 中可以存储数据的器件。一个 CPU 有多个寄存器。详细内容见第二章）

用汇编指令编写的程序称为汇编程序或汇编语言源程序，在本书简称源程序。汇编程序要比用机器指令编写的程序更容易理解和维护。

1.1.3 汇编程序

用汇编语言编写的程序大大提高了程序的可读性。可是 CPU 只能读懂机器指令，那么如何让计算机执行程序员们用汇编指令编写的程序呢？这时，就需要有一个能够将汇编指令转换成机器指令的翻译程序，它能把用汇编语言编写的源程序翻译成 CPU 能识别的机器指令序列。这里，称该程序为汇编程序。图 1.1 描述了这个工作过程。

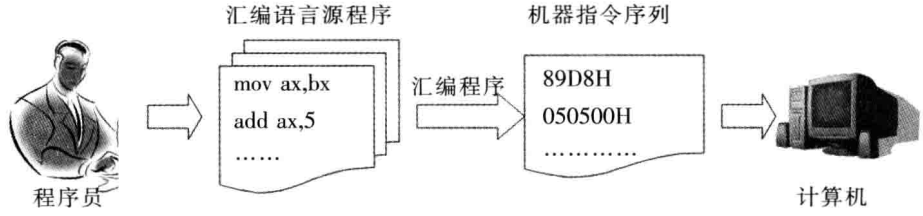

图 1.1　汇编语言指令翻译为机器指令的示意图

不难看出，汇编程序把左边汇编语言源程序翻译成右边的机器指令序列。其中把汇编语言指令“mov ax，bx”和“add ax，5”分别转换成机器指令 89D8H 和 050500H，而后者 CPU 能直接识别并执行。

目前，常用的汇编程序有 MASM、TASM 和 DEBUG 等。

1.1.4 汇编语言的主要特点

一方面，汇编语言指令是用一些具有相应含义的助忆符来表达的，所以，它要比机器语言容易掌握和运用，但另一方面，它要直接使用 CPU 的资源，相对高级程序设计语言来说，它又显得难掌握。汇编程序归纳起来大概有以下几个主要特性。

1. 与机器相关性

汇编语言指令是机器指令的一种符号表示，而不同类型的 CPU 有不同的机器指令系统，也就有不同的汇编语言，所以，汇编程序与机器有着密切的关系。由于汇编程序与机器的相关性，所以，除了同系列、不同型号 CPU 之间的汇编程序有一定程度的可移植性之外，其他不同类（小型机和微机等）CPU 之间的汇编程序是无法移植的。也就是说，汇编程序的通用性和可移植性要比高级语言程序低。

2. 执行的高效率

正因为汇编语言有“与机器相关性”的特性，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，执行速度快。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（在后续课程《编译原理》的有关章节会有详细介绍），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。而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，这也是汇编程序执行高效率的原因之一。

3. 编写程序的复杂性

汇编语言是一种面向机器的语言，其汇编指令与机器指令基本上一一对应，所以，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（如计算：A+B+C 等），就必须安排 CPU 的每步工作（如：先计算 A+B，再把 C 加到前者的结果上）。另外，在编写汇编程序时，还要考虑机器资源的限制、汇编指令的细节和限制等等。由于汇编程序要安排运算的每一个细节，这就使得编写汇编程序比较繁琐、复杂。一个简单的计算公式或计算方法，也要用一系列汇编指令一步一步来实现。

4. 调试的复杂性

在通常情况下，调试汇编程序要比调试高级语言程序困难，其主要原因有四：一是汇编语言指令涉及到机器资源的细节，在调试过程中，要清楚每个资源的变化情况。二是程序员在编写汇编程序时，为了提高资源的利用率，可以使用各种实现技巧，而这些技巧完全有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，还要清楚它在整个解题过程中的作用。三是高级语言程序几乎不显式地使用“转移语句”，但汇编程序要用到大量的、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。如果在汇编程序中也强调不使用“转移指令”，那么，汇编程序就会变成功能单调的顺序程序，这显然是不现实的。四是调试工具落后。高级语言程序可以在源程序级进行符号跟踪，而汇编程序只能跟踪机器指令。不过，现在这方面也有所改善，CV（CodeView）、TD（Turbo Debug）等软件也可在源程序级进行符号跟踪了。

1.1.5 汇编语言的使用领域

综上所说，汇编语言的特点明显，其诱人的优点直接导致其严重的缺点，其“与机器相关”和“执行的高效率”导致其可移植性差和调试难。所以，我们在选用汇编语言时要根据实际的应用环境，尽可能避免其缺点对整个应用系统的影响。

下面简单列举几个领域以示说明，但不要把它们绝对化。

1. 适用的领域

- 要求执行效率高、反应快的领域，如：操作系统内核、工业控制、实时系统等；
- 系统性能的瓶颈，或频繁被使用子程序或程序段；
- 与硬件资源密切相关的软件开发，如：设备驱动程序等；
- 受存储容量限制的应用领域，如：家用电器的计算机控制功能等；
- 没有适当的高级语言开发环境。

2. 不宜使用的领域

- 大型软件的整体开发；
- 没有特殊要求的一般应用系统的开发等。

1.2 计算机的主要性能指标

计算机是一个综合系统，因此很难用一两项具体指标来衡量其优劣。通常根据计算机的机器字长、速度、主频大小、内存的容量、内存的存取周期、可靠性及带宽等指标来衡量一台计算机的性能。

1. 机器字长

机器字长是指该计算机能进行多少位二进制数的并行运算，实际上是指该计算机中的运算器有多少位。通常计算机的数据总线和寄存器的位数与机器字长一致。如果机器字长16位，表示该机器中，每次能完成2个16位二进制数的运算。由于参加运算的操作数和运算结果既可存放在处理器内部的寄存器中，也可存放在主存储器中，因此机器字长既是运算器的长度，也是寄存器的长度。一般情况下，它也是内存的字长。通常机器字长越长，计算机的运算能力越强，其运算精度也越高。

衡量机器字长的单位可用“位（bit）”。位是计算机内部最小的信息单位，8位构成一个“字节（byte）”。现代计算机的机器字长一般都是8位的整数倍，如8位、16位、32位、64位和128位等，即字长由1个字节、2个字节、4个字节、8个字节或16个字节组成，所以也可用字节来表示机器字长。

2. 速度

长期以来，“速度”被认为是评价计算机性能的重要指标之一。使用计算机的人希望计算机的运算速度越快越好，这是无可非议的。但是应该如何正确地描述计算机的运算速度，也是一个值得探讨的问题。

CPU速度是指单位时间（秒）内能执行的指令的条数。速度的计算单位元不一，若以单字长定点指令的平均执行时间计算，用MIPS（Million Instructions Per Second）作为

单位；若以单字长浮点指令的平均时间计算，则用 MFLOPS（Million Floating Instructions Per Second）表示。现在，采用计算机中各种指令的平均执行时间和相应的指令运行权重的加权平均法求出等效速度作为计算机运算速度的标准。

3. 主频

主频又称为主时钟频率，是指 CPU 在单位时间（秒）内产生的时钟脉冲数，以 MHz（兆赫兹）为单位。如 486 DX/66 的主频为 66MHz，Pentium Ⅱ/350 的主频为 350MHz，Pentium Ⅲ/550 的主频为 550MHz 等。计算机 CPU 的时钟频率越高，运算速度越快，尤其是对于机器结构相同或相近的计算机。主频可以用来比较运算速度的高低。

4. 存储器的容量

存储器容量的大小不仅影响着存储程序和数据的多少，而且也影响着运行这些程序的速度。这是人们在购买计算机时关心的又一个关键问题。

内存储器中能够存储的总字节数称为内存（一般指 RAM）的容量。度量计算机内存的单位有 B（字节）、KB（千字节）、MB（兆字节）、GB（千兆字节）等。对于外存储器，除了以上度量单位还用到了 TB（兆兆字节）。它们之间的换算关系为：

Bit	Byte	KiloByte	MegaByte	GigaByte	TeraByte
位 →	字节 →	千字节 →	兆字节 →	千兆字节 →	兆兆字节
	8bit	1024B	1024KB	1024MB	1024GB
		2^{10}B	2^{20}B	2^{30}B	2^{40}B

现代计算机的内存容量已达到数 GB。

由于存储器种类很多，所以关心存储器的容量也不限于内存的大小，寄存器、高速缓存的大小，还有磁盘、光盘、磁带的容量，以及分散在显示卡、图形卡、视频卡、网络卡等上面的存储器容量都需要关心。

另外，对于磁盘存储器，除考虑它的存储容量外，还要考虑一些特殊的指标，如平均寻道时间、平均等待时间和数据传输速率等。所谓平均寻道时间是指磁头沿着盘径移动到需要读写的那个磁道花费的平均时间。所谓平均等待时间是指需要读写的扇区旋转到磁头下面花费的平均时间。所谓数据传输速率是指磁头找到所需要读写的扇区后，每秒钟可以读出或写入磁盘的字节数。

5. 存取周期

存储器完成一次数据的读（取）或写（存）操作数所需要的时间称为存储器的存取（或访问）时间。存储器执行一次完整的读/写操作所需要的时间称为存取周期，即从存储器中连续取（读）或存（写）两个字所用的最小时间间隔。内存大都由大规模集成电路制成，其存取周期目前为几十纳秒（ns）。

6. 带宽

计算机的数据传输率还常用带宽表示，它反映了计算机的通信能力。当然，与通信相关的设备、线路都有带宽指标。

数据传输单位是 bps，习惯缩写用 b 表示 bit，因此，bps 代表每秒传输一位或一比特（bits per second）。由于 bit 太小，所以常用 kbps 表示每秒传输一千比特，Mbps 表示每秒一兆比特，Gbps 表示每秒一吉比特。例如网络卡的速率为 10Mbps～100Mbps，调制解调

器的速率为 56kbps 等。

理想的计算机系统要求功能和程序行为之间有良好的匹配。由于计算机性能受到多个方面的影响，如程序的算法设计、数据结构、所使用的编程语言的效率、编写程序的程序员的水平以及编译技术等，所以机器性能将随着程序而变化。这说明在实际应用中要达到峰值性能是不可能的；另一方面，又不能说明机器具有某种平均性能。实际上，所有性能指标和测定结果都是执行综合性的程序产生的，因此只能在一定范围内或按调和分布来描述性能。

1.3 数据的表示和类型

在用汇编语言进行程序设计时，程序员可以直接访问内存，对数据在存储器内的表示形式要有一个清晰的认识。下面，我们只简单介绍本课程所要用到的数据表示知识，为后面的学习作一点必要的准备。

有关“数据表示”的详细内容请参阅“计算机组成原理”课程中的相关章节。

1.3.1 数值数据的表示

1. 二进制

在计算机内，数值是用二进制来表示的，它只包括 0 和 1 两个数码，很容易用电子元件的两种不同的状态来表示。例如，用高电平表示 1，用低电平表示 0。所以，计算机中通常采用二进制数。二进制数的计数特征：逢二进一，运算简单。每个二进制数按权相加就可得到其十进制数值。在书写二进制时，为了区别，在数据后面紧跟一个字母 B。

二进制的一般表现形式为：$b_{n-1}\cdots b_1 b_0 B$，其代表数值：$b_{n-1}2^{n-1}+\cdots+b_1 2^1+b_0 2^0$。

数据的二进制表示形式简单、明了，但它书写起来比较长，所以，通常情况下，我们在程序中不直接用二进制来书写具体的数值，而改用八进制、十进制或十六进制。

2. 八进制

八进制是一种二进制的变形，三位二进制可变为一位八进制，反之亦然。八进制的表示元素是：0、1、…、7。在书写时，为了区别，在数据后面紧跟一个字母 Q。如：1234Q、7654Q、54Q 等都是八进制。

八进制数在程序中的使用频率不高。

3. 十进制

十进制是我们最熟悉的一种数据表示形式，它的基本元素是：0、1、…、9。在书写时，为了区别，在数据后面紧跟一个字母 D。在程序中经常用十进制来表示数据。

4. 十六进制

十六进制是另一种二进制的变形，四位二进制可变为一位十六进制，反之亦然。十六进制的基本元素是：0、1、…、9、A、B、…、F（字母小写也可以）。其中：字母 A、B、…、F 依次代表 10、11、…、15。

在书写时，为了区别，在数据后面紧跟一个字母 H。当十六进制数的第一个字符是字