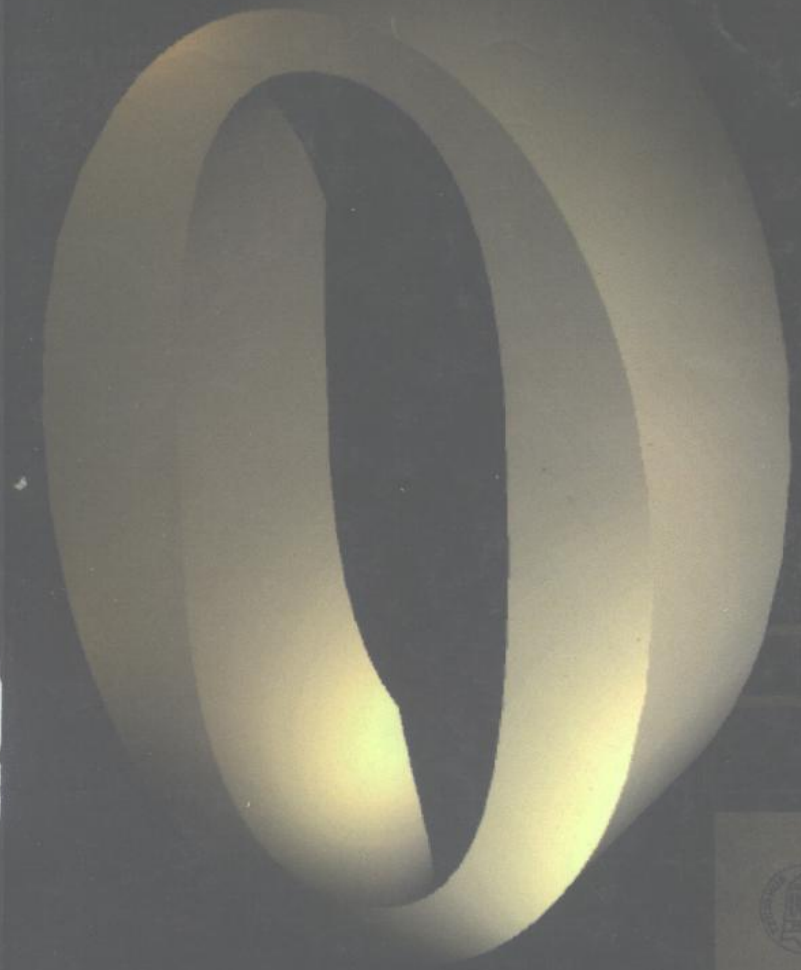
80X86

汇编语言程序设计教程

杨季文　等编著
钱培德　　审



清华大学出版社
http://www.tup.tsinghua.edu.cn

80x86 汇编语言程序设计教程

杨季文等 编著
钱培德 审

清华大学出版社

(京)新登字 158 号

内 容 提 要

本书分为三部分。第一部分是基础部分，以 8086/8088 为背景，以 DOS 和 PC 兼容机为软硬件平台，以 MASM 和 TASM 为汇编器，介绍汇编语言的有关概念，讲解汇编语言程序设计技术。第二部分是提高部分，以 80386 为背景，以新一代微处理器 Pentium 为目标，细致和通俗地介绍了保护方式下的有关概念，系统和详细地讲解了保护方式下的编程技术，真实和生动地展示了保护方式下的编程细节。第三部分是上机实验指导。

本书的第一部分适合初学者，可作为学习汇编语言程序设计的教材。本书的第二部分适合已基本掌握 8086/8088 汇编语言的程序员，可作为学习保护方式编程技术的教材或参考书，也可作为其他人员了解高档微处理器和保护方式编程技术的参考书，还可作为程序员透彻地了解 Windows 程序设计技术的参考书。

图书在版编目(CIP)数据

80x86 汇编语言程序设计教程/杨季文等编著. —北京：清华大学出版社，1998.4

ISBN 7-302-02901-6

Ⅰ.80… Ⅱ.杨… Ⅲ.汇编语言-程序设计-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(98)第 07169 号

出版：清华大学出版社(北京清华大学校内，邮编 100084)

因特网地址：www.tup.tsinghua.edu.cn

印刷：丰华印刷厂

发行：新华书店总店北京科技发行所

开本：787×1092 1/16 **印张**：38 **字数**：902 千字

版次：1998 年 6 月第 1 版 1998 年 6 月第 1 次印刷

书号：ISBN 7-302-02901-6/TP・1535

印数：0001～6000

定价：39.00 元

前　言

汇编语言面向机器，只有它能够为程序员提供最直接操纵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。该课程对于训练学生掌握程序设计技术，熟悉上机操作和程序调试技术都有重要作用。此外，“汇编语言程序设计”也是其他相关专业的必修或选修课。

目前，国内最广泛使用的 PC 系列机（包括兼容机），都以 Intel 的 80x86 系列微处理器或者兼容的微处理器为 CPU。在 Intel 的 80x86 家族中，16 位的 8086/8088 是基础，实现了以分段方式管理存储器；32 位的 80386 是高档微处理器的里程碑，实现了支持多任务的保护工作方式；基于 MMX 技术的 Pentium 是新一代的微处理器，实现了对多媒体处理的支持。本书以 8086/8088 为基础，以 80386 为重点，面向 Pentium 等新一代微处理器，讲解汇编语言程序设计的一般概念、基本技术和常用技巧，介绍宏和模块化程序设计的技术方法，讲解保护方式编程的相关概念、编程技术及实现细节。

本书分三个部分，共 12 章。第一部分是基础部分。第 1 章介绍汇编语言的特点和其他基本概念。第 2、第 3 章以 8086/8088 为背景，简要介绍 8086/8088 寻址方式、指令系统和汇编语言的常用伪指令语句后，讲解如何利用汇编语言实现程序的基本结构。第 4 章详细讲解了子程序的设计和如何调用 DOS 提供的子程序。第 5 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 BIOS 程序和调用 BIOS 程序。第 6 章以 DOS 为软件平台，讲解如何利用汇编语言编写小型应用程序。第 7 章以 MASM 和 TASM 为汇编器，介绍宏和条件汇编等汇编语言的高级技术。第 8 章介绍模块化程序设计技术以及与高级语言的混合编程。第二部分是提高部分。第 9 章介绍实方式下的 80386 及其编程。第 10 章讲解保护方式下的 80386 及其编程，该章内容十分丰富。第 11 章介绍 80486 和 Pentium 程序设计基础。第三部分是上机实验指导，安排为第 12 章，应在上机实验前先阅读了解该章内容。

本书的第一部分适合初学者，可作为学习汇编语言程序设计的教材。本书的第二部分适合已基本掌握 8086/8088 汇编语言的程序员，可作为学习保护方式编程技术的教材或参考书，也可作为其他人员了解高档微处理器和保护方式编程技术细节的参考书，还可作为程序员透彻地了解 Windows 程序设计技术的参考书。

杨季文编写第 1 章、第 3 章、第 4 章至第 7 章、第 9 章至第 11 章，朱巧明编写第 2 章，吕强编写第 8 章，曹培培编写第 12 章。由杨季文最后统一定稿。

本书从初稿到定稿的全过程都始终得到了指导老师钱培德教授的热情关心和大力支持，承蒙他审阅了全书，特在此表示衷心感谢。本书在编著过程中得到了同事鲁征山、陈时

飚、李培峰和李廷彦等同志的帮助，还得到了赵雷和朱楠灏等同志的帮助，在此表示感谢。刘文杰和许晨等同志在讲课时使用过本书的初稿，并提出了宝贵意见，在此表示感谢。

书中不妥和谬误之处难免，恳请读者批评指正。

编　者

1998 年 2 月

目　录

第一部分　基础部分

第二部分 提 高 部 分

第三部分 上机实验指导

第一部分 基础部分

第1章 绪 论

本章先介绍汇编语言的一些基本概念，然后介绍数据的表示和类型，最后简单介绍了Intel的x86家族历代微处理。

1.1 汇编语言概述

尽管在使用汇编语言进行程序设计之前，完全理解汇编语言的特点有困难，但了解汇编语言的特点对学习汇编语言程序设计是有益的。本节先说明汇编语言的内容，再介绍汇编语言的特点和使用汇编语言的场合。

1.1.1 汇编语言

1. 机器语言

CPU能直接识别并遵照执行的指令称为机器指令。机器指令在形式上表现为二进制编码。机器指令一般由操作码和操作数两部分构成，操作码在前，操作数在后。操作码指出要进行的操作或运算，如加、减、传送等。操作数指出参与操作或运算的对象，也指出操作或运算结果存放的位置，如CPU的寄存器、存储单元和数据等。

机器指令与CPU有着密切的关系。通常，CPU的种类不同，对应的机器指令也就不同。不同型号CPU的指令集往往有较大的差异。但同一个系列CPU的指令集常常具有良好的向上兼容性，也即下一代CPU的指令集是上一代CPU指令集的超集。例如，Intel 80386指令集包含了8086指令集。

机器语言是用二进制编码的机器指令的集合及一组使用机器指令的规则。它是CPU能直接识别的唯一语言。只有用机器语言描述的程序，CPU才能直接执行。用机器语言描述的程序称为目的程序或目标程序。

为了阅读和书写方便，常用十六进制形式或八进制形式表示二进制编码。例如，我们用Intel 8086指令写一个两数相加的程序片段。具体要求是把偏移2200H存储单元中的数与偏移2201H存储单元中的数相加，将它们的和送入偏移2202H存储单元。完成这一工作的程序片段包含三条机器指令，用十六进制形式表示如下：

```
A0 00 20
02 06 01 20
```

几乎没有人能直接看出该程序片段的功能，原因是程序员难以掌握机器语言。因此，程序员难以用机器语言写程序，更难写出健壮的程序；用机器语言编制出的程序也不易为人们理解、记忆和交流。所以，只是在早期或不得已时才用机器语言写程序，现在几乎没有人用机器语言写程序了。机器语言有如下缺点：机器语言不能用人们熟悉的形式来描述计算机要执行的任务；用机器语言编写程序十分繁难，极易出错；一旦有错，也很难发现，也即调试困难。

2. 汇编语言

为了克服机器语言的上述缺点，人们采用便于记忆、并能描述指令功能的符号来表示指令的操作码。这些符号被称为指令助记符。助记符一般是说明指令功能的英语词汇或者词汇的缩写。同时也用符号表示操作数，如 CPU 的寄存器、存储单元地址等。

用指令助记符、地址符号等符号表示的指令称为汇编格式指令。

汇编语言是汇编格式指令、伪指令的集合及其表示、使用这些指令的一组规则。伪指令的概念留待以后介绍。用汇编语言书写的程序称为汇编语言程序，或称为汇编语言源程序，或简称为源程序。

利用汇编语言，上述两数相加的程序片段可表示如下：

```
MOV     AL,VAR1
ADD     AL,VAR2
MOV     VAR3,AL
```

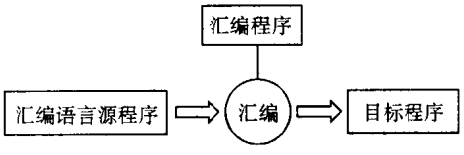


图 1.1　汇编过程示意图

显然，汇编格式指令比二进制编码的机器指令要容易掌握得多，用汇编语言编写的程序要比用机器语言编写的程序容易理解、调试和维护

3. 汇编程序

由于 CPU 能直接识别的唯一语言是机器语言，所以用汇编语言编写的源程序必须被翻译成用机器语言表示的目标程序后才能由 CPU 执行。把汇编语言源程序翻译成目标程序的过程称为汇编。完成汇编任务的程序叫做汇编程序。汇编过程如图 1.1 所示。

1.1.2　汇编语言的特点

由于汇编语言使用指令助记符和符号地址，所以它要比机器语言容易掌握得多。与高级语言相比较，汇编语言有如下特点。

1. 汇编语言与机器关系密切

因为汇编格式指令是机器指令的符号表示，所以汇编格式指令与机器有着密切的关系，因此汇编语言也与机器有着密切的关系，确切地说汇编语言与机器所带的 CPU 有着十分密切的关系。对于各种不同类型的 CPU，要使用各种不同的汇编语言。于是，对于各种不同类型的 CPU，也就有各种不同的汇编程序。

由于汇编语言与机器关系十分密切，所以汇编语言源程序与高级语言源程序相比，它

的通用性和可移植性要差得多。但通过汇编语言可最直接和最有效地控制机器，这常常是大多数高级语言难以做到的。

2. 汇编语言程序效率高

用汇编语言编写的源程序在汇编后所得的目标程序效率高。这种目标程序的高效率反映在时间和空间两个方面：其一是运行速度快；其二是目标程序短。在采用相同算法的前提下，任何高级语言程序在这两方面的效率都不如汇编语言程序，许多情况下更是远远不及。

汇编语言程序能获得"时空"高效率的主要原因是：构成汇编语言主体的汇编格式指令是机器指令的符号表示，每一条汇编格式指令都是所对应的某条机器指令的"化身"；另一个重要原因是汇编语言程序能直接并充分利用机器硬件系统的许多特性。高级语言程序在上述两点上要逊色得多。

3. 编写汇编语言源程序繁琐

编写汇编语言源程序要比编写高级语言源程序繁琐得多。汇编语言是面向机器的语言，高级语言是面向过程或面向目标、对象的语言。如下两点突出表现了汇编语言的这一特性：

作为机器指令符号化的每一条汇编格式指令所能完成的操作极为有限。例如，Z80 指令集中没有乘法指令，8086 指令集中没有能够同时完成两次算术运算的指令。

程序员在利用汇编语言编写程序时，必须考虑包括寄存器、存储单元和寻址方式在内的几乎所有细节问题。例如：指令执行对标志的影响，堆栈设置的位置等。在使用高级语言编写程序时，程序员不会遇到这些琐碎却重要的问题。

4. 汇编语言程序调试困难

调试汇编语言程序往往要比调试高级语言程序困难。汇编格式指令的功能有限和程序员要注意太多的细节问题是造成这种困难的两个客观原因；汇编语言提供给了程序员最大的"舞台"，而程序员往往为了追求"时空"上的高效率而不顾程序的结构，这是造成调试困难的主观原因。

1.1.3 恰当地使用汇编语言

1. 汇编语言的优缺点

为了恰当地使用汇编语言，我们先明确一下它的优缺点。

汇编语言的主要优点是利用它可能编写出在"时空"两个方面最有效率的程序。另外，通过它可最直接和最有效地操纵机器硬件系统。

汇编语言的主要缺点是它面向机器，与机器关系密切，它要求程序员比较熟悉机器硬件系统，要考虑许多细节问题，最终导致程序员编写程序繁琐；调试程序困难；维护、交流和移植程序更困难。

正是由于汇编语言与机器关系密切，才使汇编语言具有其他高级语言所不具备的上述优点和缺点。为了利用汇编语言的优点，必须付出相应的代价。但汇编语言的每一个优点常常闪耀出诱人的光芒，使人们勇敢地面对它的缺点。

2. 使用汇编语言的场合

根据汇编语言的优缺点，我们要恰当地使用汇编语言，即尽可能地“扬长避短”。是否利用汇编语言编写程序，要看具体的应用场合，要充分考虑到软件的开发时间和软件的质量等诸多方面的因素。我们认为下列应用场合，可考虑使用汇编语言编写程序。

(1) 对软件的执行时间或存储容量有较高要求的场合。例如：系统程序的关键核心，智能化仪器仪表的控制系统，实时控制系统等。

(2) 需要提高大型软件性能的场合。通常把大型软件中执行频率高的子程序(过程)用汇编语言编写，然后把它们与其他程序一起连接。

(3) 软件与硬件关系密切，软件要有直接和有效控制硬件的场合。如设备驱动程序等。

(4) 没有合适的高级语言的场合。

3. 适度地追求“时空”效率

在用汇编语言编写程序时，追求“时空”效率要适度。在编写汇编语言程序时，我们要尽量利用最恰当的指令，以便节约一个字节或节省几个机器周期。但时至今日，计算机硬件系统的整体性能已极大地提高，所以，除非不得已，不要为节约少量字节或机器周期而影响程序的结构性、健壮性和可读性等。要在确保汇编语言程序上述性能良好的前提下追求“时空”性能。

1.2 数据的表示和类型

熟悉数据在计算机内的表示形式是掌握汇编语言程序设计的关键之一。本节简单介绍数据的表示形式和类型。

计算机中存储信息的最小单位称为位，在绝大多数系统中它只能表示两种状态。这两种状态可分别代表 0 和 1。计算机系统内部采用二进制表示数值数据，也采用二进制编码表示非数值数据和指令，其主要原因就在于此。

1.2.1 数值数据的表示

所谓数值数据就是数。这里仅介绍定点整数的有关内容。

1. 数的二进制表示

尽管日常生活中大多采用十进制计数，但在计算机内，数却大多采用二进制表示。某个二进制数 $b_nb_{n-1}\cdots b_2b_1b_0$ 所表示的数值用十进制数来衡量时，可利用如下按权相加的方法计算得到：

$$b_n2^n+b_{n-1}2^{n-1}+\cdots+b_22^2+b_12^1+b_02^0$$

在书写时，为了与十进制数相区别，通常在二进制数后加一个字母 B。

2. 有符号数的补码表示

为了方便地表示负数和容易地实现减法操作，有符号数采用补码形式表示。所以，有符号数二进制表示的最高位是符号位，0 表示正数，1 表示负数。正数数值的补码形式用二进制表示。为得到一个负数数值的补码形式，方法可以是先得出该负数所对应正数的二进