

C/C++程序 初学者指南

Your First C/C++ Program

- 初学者的良师益友
- 循序渐进、由浅入深
- 丰富的例题，边学边实践

Alan R. Neibauer 著

胡海天 胡希平 译

胡希平 审校

電子工業出版社



Your First C/C++ Program

C/C++程序初学者指南

[美] Alan R. Neibauer 著

胡海天 胡希平 译

胡希平 审校

電子工業出版社

内 容 提 要

本书是学习C/C++的一本入门书籍。全书分十三章，包括C/C++程序的所有内容和方面，从基本概念到输入、输出语句和条件、循环；从函数、数组到指针和结构体，以及磁盘文件的读写和如何使用打印机等等，最后一章给出一个综合程序实例。每章末附有思考题和习题。

本书的特点是图文并茂，通俗易懂。内容安排循序渐进，示例丰富，概念清晰，重点突出。

对计算机程序一无所知的读者，本书是一本难得的自学教材。对初、中级程序员、大专院校师生和计算机培训班学员，本书也可作为学习参考和培训教材使用。



Copyright©1995 SYBEX Inc., 2021 Challenger Drive, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版，SYBEX公司已将中文版独家版权授予北京美迪亚电子信息有限公司。未经许可，不得以任何形式和手段复制或抄袭本书内容。

C/C++程序初学者指南

[美] Alan R. Neibauer 著

胡海天 胡希平 译

胡希平 审校

责任编辑 李 湘

*

电子工业出版社出版（北京市万寿路）

电子工业出版社发行 各地新华书店经销

北京美迪亚电子信息有限公司排版

北京顺义颖华印刷厂印刷

*

开本：787×1092毫米 1/16 印张：17.5 字数：450 千字

1996 年 1 月第 1 版 1996 年 1 月第 1 次印刷

印数：5000 册 定价：35.00 元

ISBN 7-5053-3354-2/TP·1287

著作权合同登记章

图字：01-95-599

出版说明

计算机科学技术日新月异。为了引进国外最新计算机技术，提高我国计算机应用与开发的水平，中国电子工业出版社与美国万国图文有限公司合资兴办的北京美迪亚电子信息有限公司取得了美国SYBEX公司的独家版权代理。SYBEX公司授权本公司通过电子工业出版社等出版机构全权负责在中国大陆出版该公司的中文版和英文版图书。现在与广大读者见面的是最近推出的第一批图书。今后我们还将陆续推出SYBEX公司的最新计算机图书和软件，为广大读者提供更好的服务，传递更多的信息。

美国SYBEX公司是世界著名的计算机图书出版商，该公司自1976年创办开始，其宗旨就是通过出版有效的，高质量的图书向计算机用户介绍实用技巧。我们优选翻译出版的图书是SYBEX公司的最新计算机图书，并采用了该公司提供的电子排版文件，从而提高质量并大大缩短了图书的出版时间，从根本上改变了以往翻译图书要落后原版书较长的“时差”现象，这在电子技术日新月异的时代具有深远意义。

北京美迪亚电子信息有限公司

1995年10月

译者序

C是一种编译语言，C++是C的超集。由于C具有快速性、可移植性和可控性，它目前已成为超过其它计算机语言的首选语言，得到越来越广泛的应用。

随着计算机应用的广泛深入和家用电脑的日益普及，人们已不满足于使用现成的字处理软件、电子表格软件、数据库软件、游戏软件等各种应用软件，而欲窥视程序内部的机理，萌生了学习编写程序的念头。对于一个从未接触过计算机程序的人来说，在充斥于市的计算机图书中寻觅一本合适的人门书籍，并非如想象那样简单。尽管许多C程序语言的书籍是针对初学者的，但专业味仍较浓，它们要求读者具备语言、算法、数据结构和程序设计方法等方面的知识，或者要有掌握一种其它高级语言的经验。SYBEX公司针对这一大批读者的需求，组织出版了本书——一本真正适合初学者的入门书籍，填补了计算机图书市场上的一个空缺。

读者阅读本书可以发现，作者运用贴近日常生活的语言，配合许多插图来阐述概念，还有丰富的示例程序，再加上附带有包含C编译器、程序编辑器和全部示例程序的软盘，使原来枯燥无味的程序语言学习变成了饶有兴味的边学边实践的过程。即使对C/C++程序语言已有基础的人，阅读本书仍可得到许多启迪。

本书作者毕业于宾夕法尼亚大学，目前正在教授大学水平的程序语言，并兼顾写作、咨询。他为SYBEX公司撰写了许多计算机和软件方面的普及书籍。本书是他1995年推出的新作。

在本书翻译过程中，自始至终得到张喜乐和李莱高级工程师的帮助和支持，还承蒙刘意青教授和张中华老师对译稿提出了不少宝贵意见，林莉帮助录入书稿，在此谨表示衷心的感谢。

由于时间紧迫，加之译者水平所限，书中疏漏和不当之处在所难免，敬希读者和专家批评指正。

译者

1995年仲夏于北京

致 谢

本书面世归功于许多有才能的人的帮助和支持。组稿编辑Dianne King和作稿编辑Gary Masters培植了这个项目。Gary慧眼识珠，一开始就看到了本书的价值，他帮助组织本项目的实施。编审Jim Compton一丝不苟地审阅并加工润色书稿，在此谨表示衷心感谢。

我也要感谢C WARE公司的好心人，他们不仅慷慨同意本书包含PCC编译器，而且特地制作了一个经济上负担得起的完善的编译工具。

在出版事务方面，十分感谢技术编辑Erik Ingenito，录入员Lisa Jaffe，校对员Janet Boone，索引员Ted Laux的辛勤工作。另外，对Veronica Eddy和Jamie Wright提出封里式样，设计师Alissa Feinberg和画家Ingrid Owen把这个式样变成现实的努力，表示感谢。还要感谢Rudolph Langer博士和其他对本书作出贡献的SYBEX公司同仁。

最后，我要诚挚感谢Barbara Neibauer。她是我书稿和插图的第一个读者，她不仅安排我的生活，而且为了本书，她学习了C和C++程序，多么可爱的妻子！

07/5/12/47

序 言

把你听说的关于编制计算机程序是多么困难的故事抛诸脑后。不要理会那些挣扎和痛苦的传说。即使你不是数学博士或理科博士，也不必担忧。

学习如何编写计算机程序，以及实际编写程序是非常有趣的事。如果你善于逻辑思维，如果你喜欢猜谜游戏和解小测试题，或者你想更好地控制你的计算机，那么你就是非常适合编程的人。

你现在已经拥有最好的入门书籍。本书是专为初学者、刚刚开始的人设计的。它实际上假定你对计算机程序一无所知。(本书对那些从其它语言，如BASIC、Pascal以及诸如WordPerfect、Lotus和Excel等软件的宏语言转向C和C++的读者，也是非常适合的。)

事实上，你学习编程所需要的一切，就是本书和你的求知欲。配合本书的软盘还包括一个功能强大的C编译器和一个用来书写你自己程序的编辑器。

《C/C++程序初学者指南》着重讨论由K&R和ANSI C标准定义的C版本，以及C++的基本特性。你在本书中学到的所有技术都可以应用于C和C++程序。如果你对C++感兴趣，注意阅读与这个C语言的超集有关的专门的节和注释。

关于本书的内容

有了本书，你将循序渐进地学习如何编写程序。你会发现由丰富的插图和示例配合的讲解，清晰易懂。通过研读这些讲解，参看插图和程序清单，你将以适当的顺序学习编程的每个方面。每章末都附有思考题和习题。

尽管本书是为初学者设计的，但它的内容是详尽的，没有遗漏任何东西。你不仅能学习一种程序语言，而且还可以学到程序设计和解题的逻辑方法。

你将从第一章关于编程过程和第二章关于C和C++程序的结构和特点开始学习。在第三章中，你将学习如何运用变量和常量与程序对话。

接下来，你在第四章中学习如何在监视器上显示信息，在第五章中学习如何从键盘输入信息。在第六章中，你学习如何用运算符书写实现算法的程序。

第七章有关函数，你会发现如何把一个程序分为易于处理的块，分段建立你的程序。第八章讨论做决定的问题，第九章告诉你如何使用循环来重复指令。

随着你的编程技巧的提高，你在第十章中学习数组和字符串，第十一章学习有关指针和结构体。第十二章中，你学习如何读写磁盘文件，以及如何访问你的打印机。

为了把你的技术组合起来，第十三章介绍一个综合性程序示例，用于建立一个数据库应用程序。通过阅读该章和其中的程序清单，你会学到建立专业应用程序的技巧。

本书的附录解释如何安装和使用在随书销售的磁盘上的编译器和编辑器，以及如何访问示例程序和每章末习题的答案。

阅读第一章和第二章之后，遵照附录的指示安装磁盘上的程序。然后你就可以接着阅读下面的章节。说不定你很快就会着手自己编写用于办公室、学校和家庭的程序。

完备的编程工具

如果你已经有一个你自己的C或C++编译器，那么你可以用它来学习C和书写本书给出的程序。但是如果你没有编译器，那么你需要的全部东西都在随书销售的软盘上。

随书销售的软盘不只是包括示例程序，它还包括一个功能齐全的C编译器、一个连接器、一个完备的函数库和一份80页的参考手册（英文），你可以用字处理器软件或DOS的Print命令把该参考手册打印出来。

软盘包括一个用来书写程序和编辑程序的文本编辑器，如果你以后要学习汇编语言，甚至还有一个汇编器。你也可以找到本书列出的所有程序的拷贝。你可以按原样编译和运行这些程序，或者你可以在编辑器中把它们打开，以供学习或借鉴。

目 录

第一章 概述	1
计算机程序	1
编程语言	2
编译器	6
翻译器	7
为什么要用C/C++?	8
函数库	9
C的演变	11
C能做什么和不能做什么	13
编程的步骤	13
学习编程	15
编程需要些什么	15
C/C++展望	16
思考题	16
习题	16
第二章 这就是C/C++	17
C程序的结构	17
大写和小写	19
return()函数	19
使用注释	20
理解参数	22
#include命令	23
程序设计	24
思考题	25
习题	25
第三章 变量和常量	26
字符型数据	26
整数	27
小数	28
为什么用整数?	29
常量和变量	29
常量和变量的命名	30
常量说明	31

为什么要用常量?	35
变量说明	37
赋值	38
字符串变量说明	38
数据类型和函数	42
文字	42
计划你的程序	43
思考题	44
习题	44
第四章 C/C++的输出	45
puts()函数	45
putchar()函数	46
控制码	47
通用的printf()函数	54
选择正确的输出命令	65
C++的输出	66
程序设计	68
思考题	69
习题	70
第五章 C/C++的输入	71
gets()函数	72
getchar()函数	74
&地址运算符	76
scanf()函数	78
C++的输入	85
未赋值的变量	86
有用的输入算法	87
思考题	88
习题	89
第六章 运算符	90
算术运算符	91
运算符和数据类型	94
表达式	96
优先级别	98
有用的处理算法	102
程序设计	109
思考题	111
习题	111

第七章	如何运行函数	113
	使用函数	114
	函数中的变量	117
	传递参数	122
	返回变量	126
	使用宏	129
	程序设计	132
	思考题	133
	习题	133
第八章	让计算机做决定	134
	大if	134
	多语句	136
	if...else命令	137
	逻辑运算符	139
	嵌套if语句	142
	switch/case/default结构	144
	程序设计	147
	思考题	149
	习题	150
第九章	重复	151
	使用for循环	151
	使用do...while循环	158
	使用while循环	160
	循环类型的组合使用	162
	程序设计	162
	思考题	167
	习题	168
第十章	数组和字符串	169
	数组	169
	数组练习	180
	字符串	182
	程序设计	190
	思考题	190
	习题	191
第十一章	结构体和指针	192
	使用结构体	192
	理解指针	205
	思考题	212

习题	212
第十二章 输出到磁盘和打印机	214
理解文件	214
说明文件	216
打开文件	216
关闭文件	219
输入和输出函数	220
使用字符	221
使用文本行	223
格式化输入和输出	226
使用结构体	229
读入数组中	232
给文件增加数据	234
文本和二进制格式	234
打印数据	236
程序设计	238
思考题	238
习题	239
第十三章 编程技巧的综合应用	240
应用程序	240
全局说明	241
main()函数	242
增加一个记录: addcd()函数	244
删除一个记录: delcd()函数	246
改变CD数据: chcd()函数	248
改变位置: chloc()函数	250
显示一个记录: locate()函数	251
打印一个报告: plist()函数	252
记录排序: sort()函数	253
附录A 示例软盘的使用	255
安装你的编译器	255
使用CEDIT程序编辑器	256
使用PCC编译器	257
使用示例程序	260
词汇表	262

第一章 概 述

人们由于各种各样的理由，决定学习计算机编程。当然，根本的目的是为了钱。大多数程序的编写都是作为赚钱的手段。一些程序针对特定的用户或工业部门，另一些程序则是为满足广大市场的需求，但它们的目标是相同的——为了获得商业上的成功。虽然你可能永远也不会写出“杀手”软件，但你可以通过为他人编写程序来赚钱。

你编程可能只是为了迎接挑战 and 增加经验，就象其它人登山、做填字游戏或者参加奥林匹克队选拔赛的原因一样。如果你是这种类型的程序员，那么在完成一个程序以后，你就会获得个人的满足。每个程序都是一道猜谜题和一个智力测验，你与计算机竞争，为的是最终控制计算机。

你编程也可能为了解决一个具体问题。你需要用你的计算机执行某项任务，但找不到或买不起满足你需要的商品软件。程序可能简单得象记录烹饪菜谱，也可能复杂得象解决独一无二的商业或工程问题。

无论你为计算机编程的原因是什么，编程本身可以提供一种个人的成就感和胜利感。当程序正确工作时，你获得的满足令人陶醉。当程序不能工作时，你也会感到无比沮丧。但和其它一切挑战一样，它是值得的。

计算机程序

你也许有过一些运行计算机程序的经验——字处理程序、电子表格程序、数据库，甚至一、两个游戏软件。然而，当你运行一个程序时，你只是从外部审视程序。你没有真正看到程序正在做什么，而只能看到程序运行的结果。

当你编写程序时，你是从内部观察程序。你知道程序是如何工作的，以及程序为什么那样工作。这是观察计算机程序的一个迥然不同但又很有趣的方式。无论你对运行程序有多少经验，不管你对计算机语言有多少了解，编写程序是一个全新的挑战。

你曾给寻找地址的人指过路吗？你曾教过别人做某件事吗？那正是计算机程序所做的事情。计算机程序只不过是给你的计算机的一系列指令，它们用计算机能够理解的语言，按照十分精确的顺序告诉计算机到底做什么，如此而已。

当计算机对有些事情不理解时，它甚至会显示一个出错信息来告诉你。于是你就可以按计算机能领会的方式重写一遍指令，简单地再试一次。有时，即使你的指令有错误，计算机也能够执行！你曾有过问路后严格按照指示去做却总是到达错误地点的经历吗？当然，你的程序也会发生同样的事情。这些是最难解决的问题。因为你可能根本不知道存在错误。

现在，让我们再进一步考虑我们给程序下的定义。

指令精确地告诉计算机做些什么 “精确地”一词是个关键。每个指令必须精确，不能省略任何步骤。譬如，假如有人问你到附近邮局怎么走。你告诉他们，“走过两个街区，在弗兰克林街向左转。”你只是假定他们会看见邮局并走进去。

C++注释

C++不象C和其它编程语言那样程序化。有些指令是针对“对象”和“事件”的，而不是给出许多步骤的次序。然而，不论程序化与否，指令仍必须是明确和完整的。

对于计算机，你不能假设任何事情。因此，如果你要告诉计算机怎么找到邮局，你的指示应该非常类似于下面这样：

向北走两个街区。

向左转90°。

直走50英尺。

向左转90°。

上四个台阶。

打开门，走进去。

举例来说，当你用字处理程序存储一个文档时，程序必须精确地知道执行哪些步骤。程序必须知道在磁盘上的何处放置文档。如果存储一个新文档会覆盖你的磁盘上已存在的文档时，程序还必须具有能够显示某类警告信息的指令。

指令必须有正确的顺序 如果你告诉计算机去做A，B，然后做C，计算机就会按这个顺序去做。计算机不会想“嗨！这看起来不太对，也许我们应该先拐入弗兰克林街。”然而，你可以让计算机做决定。你的程序可以决定“如果她的薪水超过\$50,000，那么扣除25%的税，否则扣除15%。”但是，你必须告诉计算机何时做决定，以及在每种条件下做什么。

譬如，让我们再次考虑试图存储与你磁盘上已存在文档同名的新文档时的情形。如果程序删掉原先的文件，存入新文件，然后再来问这是否是你想要做的事，那么程序做得不是非常好。程序应该先显示信息，然后根据你的输入决定做什么。如果你输入OK，程序覆盖磁盘上的文档。如果不是，程序要求你输入一个新文档名。

编程语言

计算机程序的第三个要求是，它必须是计算机能够理解的语言。

在你的计算机内部有一个微处理器，它是一块集成电路，用来控制计算机中发生的每件事情。（见图1.1）

当你的程序告诉计算机在监视器上显示一个信息，或者在打印机上打印这个信息时，微处理器就发送出适当的电信号。这些信号告诉计算机在内存的何处找到信息，以及把这些信息发送到何处。微处理器做的绝大多数事情是看不见的——你将程序输进计算机，并假定微处理器知道它该怎么做。但是要了解程序是如何工作的，你应当懂一点——只需要一点点

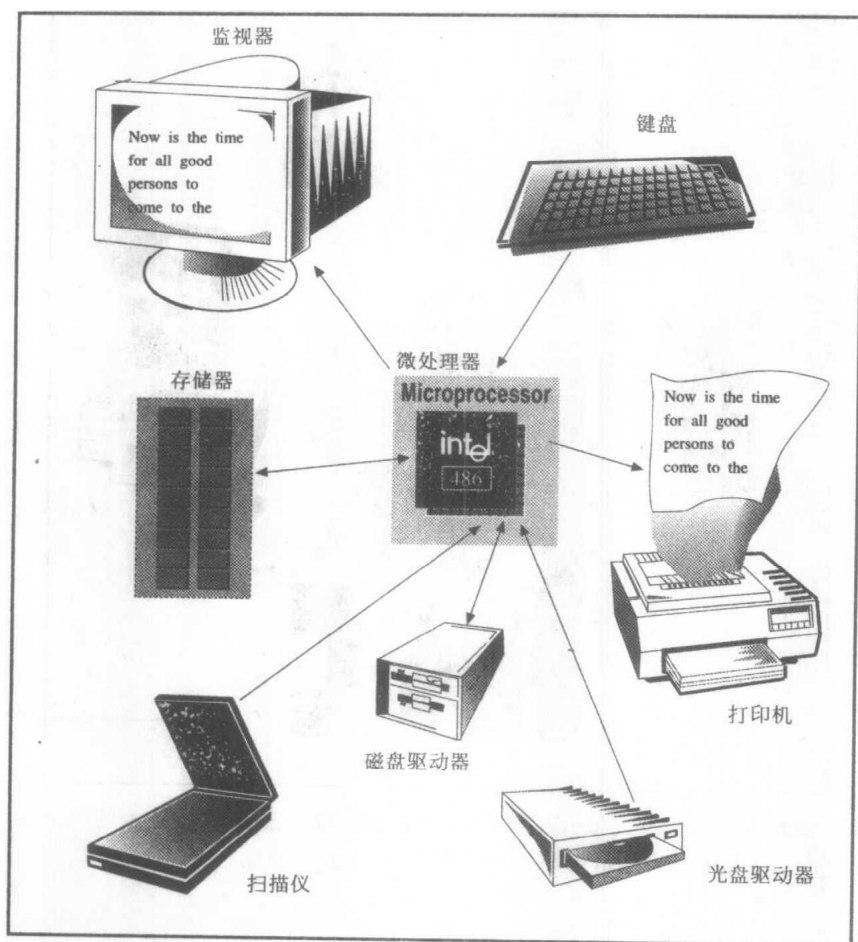


图1.1 微处理器控制计算机中的每件事情

——关于微处理器是如何工作的原理。

从技术上讲，微处理器只能执行四种操作。它可以把数据从一个内存单元传送到另一个内存单元，改变一个内存单元中的数据，确定一个内存单元是否包含特定数据，以及改变它所执行的指令的顺序。所有这些操作都是通过发送、接收或监视电脉冲来实现的。

计算机把这些脉冲处理为两种状态(电平)：一个电信号可能是开或者是关。为了完成一项任务，我们给微处理器输入一系列信号——开或关的状态。例如，图1.2所示的开与关的组合打印字符A。

为了在这个最基础的层次上写出计算机程序，我们用数字0代表关状态，1代表开状态。我们称这些为二进制数(位)，或二进制指令。它们是建立在二进制数字系统的基础上，只用0和1的组合来代表所有数字。

在计算机和计算机程序刚诞生时，程序就是通过直接操纵这些开和关信号来执行的。没有什么微处理器，技术人员作为控制器，用物理方法将一系列开关置开或置关。但是，由于要执行一个完整的程序，可能要用数以千计的单个信号，因此，编写程序是一件非常费时

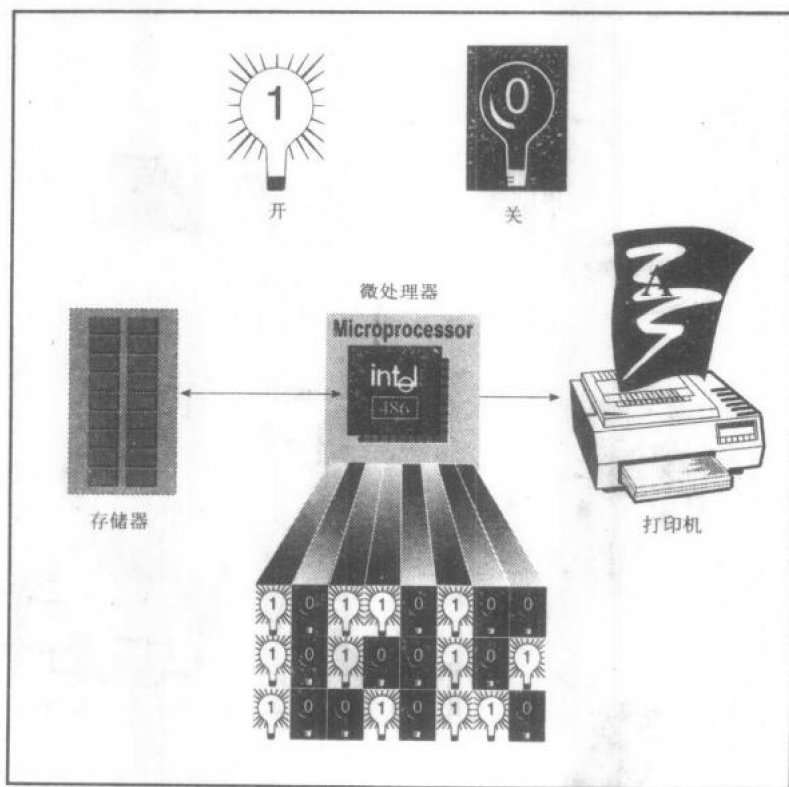


图1.2 一系列开或关电脉冲告诉微处理器执行什么任务——在本例中，是从内存中取出一个字符并把字符送到打印机

间的工作。

随着计算机的发展，将程序一次输进计算机，然后让计算机执行指令成为可能。在汇编语言产生之前，它仍需要数以千计的单个0和1。

汇编语言使用助记符来代表与硬件直接相关的计算机操作。一个助记符是易于记忆的单词或缩写，它代表一个完整的微处理器操作。例如，代码**MOV**告诉计算机将某个信息从一个内存单元传送到另一个中。代码**JMP**告诉计算机转移到内存中的另一个单元。因此，汇编语言程序员可以不用写出一系列0和1，而使用诸如**MOV**和**JMP**的代码，每个代码代表八个或八个以上的二进制指令。

如图1.3所示，一个叫做汇编器的程序将这些代码翻译成计算机能够理解的单个电信号。由于每个代码同微处理器的内部功能直接相关，因此所得到的程序可以运行得非常快。但是，汇编语言使用起来还是很费时间，并且需要大量代码。

今天，大多数程序员使用高级语言，它的指令是人们能理解的单词，而不是简单的助记符或0和1。每个单词代表一个完整的实际操作，而不是一个微处理器操作。举例来说，C函数**puts**(用于将一串字符放到监视器上)告诉计算机在屏幕上显示某个信息。同样的功能可能要用许多汇编语言助记符，也许上百个二进制电脉冲。图1.4示出在屏幕上显示一个单词的简单C/C++指令，以及功能近似的汇编语言指令和二进制代码。你觉得哪种程序语言读写

更容易?

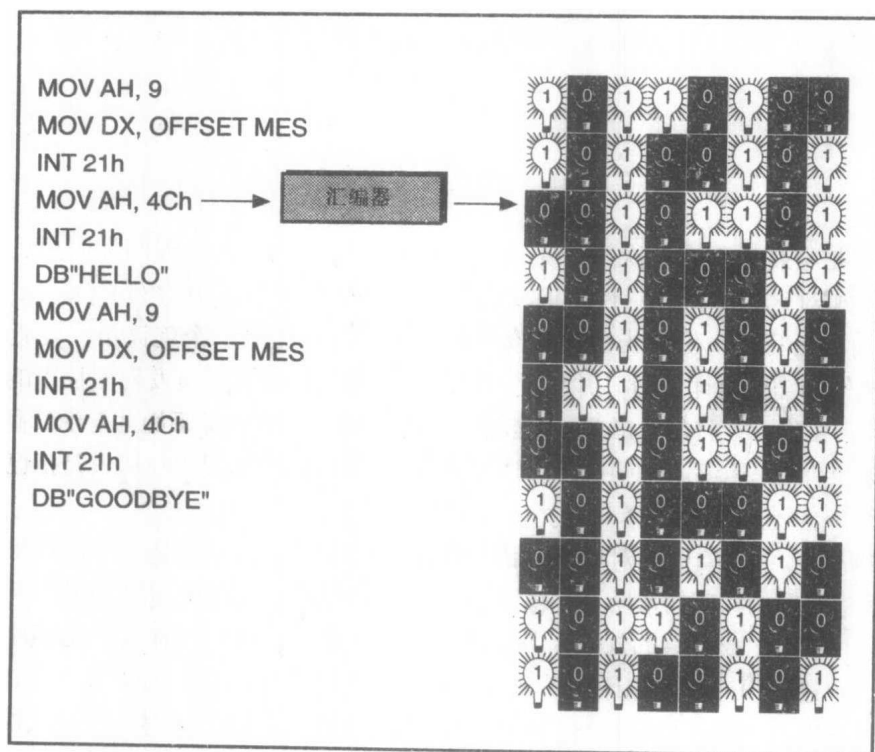


图1.3 汇编器将汇编语言指令转换成二进制代码

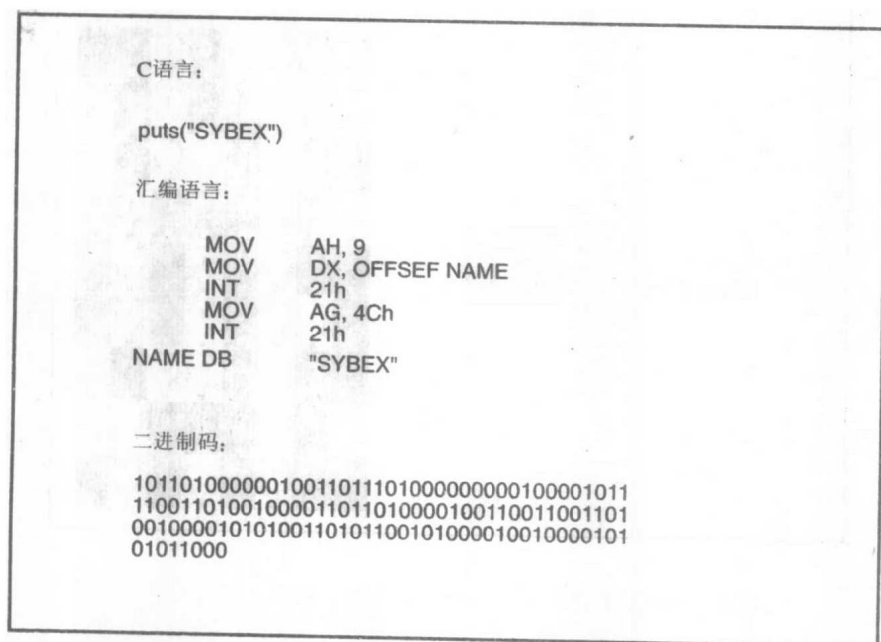


图1.4 C、汇编语言和二进制指令的比较