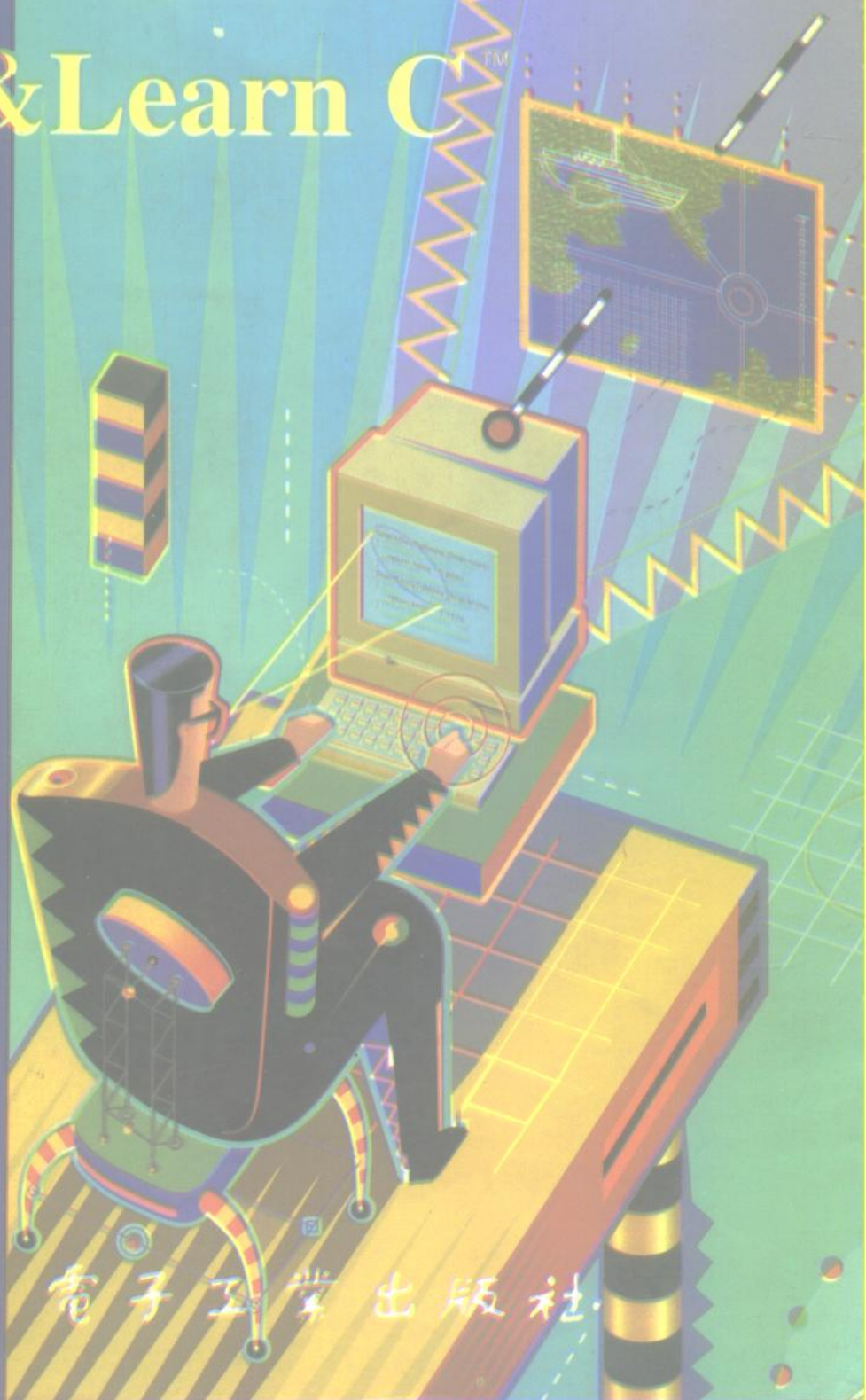


[美] Tom Swan 著 唐键 李凯 译 宋荣 审校

即学即用C语言

Type & Learn CTM

利用著名作家 Tom Swan 赞赏的
即学即用的方法迅速掌握 C 语言
学习编写 ANSI C 语言编译程序



电子工业出版社

381096

即学即用丛书

即学即用 C 语言

[美] Tom Swan 著

唐健 李凯 译

宋荣 审校



電子工業出版社

(京)新登字 055 号

©本书英文版由 IDG Books Worldwide, Inc. 出版。本书中文版由 IDG Books Worldwide, Inc. 授权由电子工业出版社出版发行。版权所有, 翻印必究。本书封面贴有激光防伪标志, 无标志者不得进入销售。

JS 109/20

即学即用丛书

即学即用 C 语言

[美] Tom Swan 著

唐健 李凯 译

宋荣 审校

责任编辑: 俞天林

电子工业出版社出版

北京海淀区万寿路 173 信箱 (100036)

电子工业出版社发行 各地新华书店经销

北京市顺义县天竺颖华印刷厂印刷

开本: 787×1092 毫米 1/16 印张: 22.25 字数: 535.4 千字

1995 年 3 月第一版 1995 年 3 月北京第一次印刷

印数: 8,000 册 定价: 45.00 元

ISBN7-5053-2767-4/TP·875

“即学即用”丛书的优点

《即学即用 C 语言》是“即学即用”系列丛书中的一本，这本书是由程序员出版社出版发行。“即学即用”丛书的编撰者们懂得：初学者在学习计算机编程时会遇到很多的困难。为了帮助初学者尽快、顺利地学会计算机编程，出版者和编程领域中的最优秀作者之一 Tom Swan 合作，提出旨在帮助初学编程者的一种学习方法。“即学即用”丛书的出版发行，则是 Tom Swan 和 IDG 图书出版公司两者广泛调查研究与很多次的试用检验的产物。

本书的宗旨很简明，即通过实践学习掌握知识。事实证明，采用实际动手操作的这种学习方法，不仅能加快学习进程；还有助于记住所学的东西。正如人们做了某事，并知道事情的结果后，一般就不会忘记自己曾做过了什么，以及为什么要做它。

专业程序员常说阅读计算机程序是件很有趣的事；而编写程序代码则更是一种享受。能否达到专业程序员的这种境界是读者成败的关键所在。本书的宗旨也说明了，为什么现在的每一本计算机书都附带有相应配套的软件。要知道，这些配套软件并不是为了凑数，可有可无；而是本书主题内容的一个关键环节，它能帮助读者立刻开始进行程序设计。

系列丛书的作者们不仅能深入浅出地为初学编程者传授深奥的知识，还有讲授程序设计方面的丰富经验。读者定能从这种常用的、要求实际操作的方法中获益。学完本书后，读者不仅能了解程序是如何运行的，而且已经练习编写了几个程序。到那时，读者的编程技巧将提高到一个新水平。

前 言

本书是适合于初学者的“即学即用”系列编程指南丛书之一。在撰写本书时，作者希望力求做到：本书具有易于掌握的特点。同时，也力求达到：不论是年轻的初学者，还是经验丰富的编程能手（“活到老，学到老”）都能借助本书学习 C 语言。

实际情况常常是，早期的编程参考书所造成的麻烦远比它们能解决的问题多得多。当开始学习用 C 语言编程时，人们总是需要一本参考书，并要求这本参考书能告诉读者在编程过程的每一步中要做什么；同时，也希望它包括读者开始学习时所需要的一切知识。

《即学即用 C 语言》一书配有 Borland 公司的世界著名软件 Turbo C++ 编译器的专用版本。读者只需安装附带磁盘中的这个软件，然后翻到第 1 章，就可以开始学习 C 语言了。

为何我们要选择 C 语言呢？这是因为 C 语言很流行，C 语言不仅适用于绝大多数计算机系统，而且相对容易学习。C 语言不是很复杂的计算机语言；它只有几个最基本的命令，因而没有必要记住许多东西。其实本书中，C 语言的主要部分只在三个章节中做了介绍。尽管 C 语言相对简单，但它却是编写计算机程序的一种能力极强的语言。事实上，世界上绝大多数软件都是用 C 语言编写的，预计在未来的几年里，这种趋势仍将保持。

只需花费一天时间掌握第一部分介绍的“即学即用”方法，读者就能学到足够多的 C 语言知识来自己编写程序了。无论是打算从事于专业软件设计的读者，还是只想更多地了解计算机软件内部是如何工作的读者，都最好首先判断一下 C 语言是否是自己正确的语言选择。

对读者而言，学会如何编程是一件激动人心的乐事；而对作者而言，如果能得知本书对读者学习编程有所帮助，那么写这本书的初衷也就实现了，作者本人也会感到万分的荣幸。

内容简介

《即学即用 C 语言》一书由四部分组成。

第一部分简介了 C 语言，并说明了如何运用“即学即用”方法。特别指出在掌握了 Borland 公司的 Turbo C++ 这个专用版本后，读者就能快速领会 C 语言的主要特征。

第二部分对 C 语言编程基础知识和技巧的方方面面进行了探讨。学完这部分内容后，读者就有能力用 C 语言开发自己的软件了。

第三部分讲述了一些 C 语言标准函数。读者可从中学习输入输出操作，字符串处理，数学函数，内存管理技术等等编程技巧。

第四部分实现了一些算法，并以此来圆满结束 C 语言的学习。这些算法包括解决诸如排序、搜索和用户自定义数据结构等各种类型的问题；而且，所有这些算法的思想都是用 C 语言的范例程序来阐明的；其中，还包括了一个演示文件处理技术的发送文件清单的数据库管理员程序。

本书的约定

为了有助于读者更好地获取信息，书中有几个有帮助作用的图符：



注释：给出本主题的重要补充信息。



警告：有助于避免灾难的发生。



技巧：提出节省时间的建议，以使程序设计更好，运行效率更高。



提示：提醒读者程序设计的另一种方案，它可能是读者未曾想到的。

“循序渐进 (Step-by-Steps)”是专为读者进行“即学即用”方法练习而设计准备的，它贯穿于全书之中。它将以如下形式出现：

循序渐进 (Step-by-Steps)

1. “循序渐进”的内容和正文的内容彼此相互独立。
2. “循序渐进”提示读者应打开存放在磁盘上的哪个程序清单。
3. 最后，“循序渐进”将说明从何处开始进行“即学即用”方法的练习。

目 录

前 言

第一部分 C 语言基础知识简介 1

第 1 章 即学即用方法 2

敲过一次，记忆永久 2

为何我们选用 C 语言？ 3

什么是编译器？ 3

什么是连接器？ 5

几个有用的新术语 6

C 语言简史 7

下一代 C 语言 8

如何安装 Torbo C++ 8

第一份 C 语言程序 9

第二份 C 语言程序 11

第 2 章 入门须知 15

系统配置 15

联机帮助 15

菜单和命令 15

目录选项 17

环境选项 18

目录命令 18

编辑器 19

编译器 20

调试器 21

单步调试 21

监视变量值 22

计算表达式值 23

设置断点 24

工程管理文件 25

第 3 章 C 语言程序设计 26

有关 main 主函数 26

C Shell 28

注释 29

变量 30

标识符 32

保留关键字 32

符号常量 32

语句	34
表达式	35
函数	36
函数参数和变元	38
数组	39
字符串	40
循环	41
好消息	43
第二部分 C语言基础和编程技巧	45
第4章 数据类型和变量	46
数据类型中有什么?	46
C语言的数据类型	46
变量	47
功能强大的转义符	49
类型修饰符	51
变量名	52
初始化变量	53
常量	54
文字常量	54
符号常量	57
实例常量	58
类型变量的大小	60
浮点数据类型	62
第5章 操作符和表达式	64
C语言的操作符	64
操作符的功能是什么?	64
算术运算符	64
一元操作符	65
增量与减量操作符	66
位操作符	68
优先级和求值顺序	70
括号	70
缺省操作优先级	71
简化操作符	72
第6章 流程控制语句和关系操作符	75
确定型关系	75
关系型操作符	75
相等操作符	75
逻辑运算符	76
逻辑非	77
条件判断	78
If-else 语句	78

Else-if 语句	83
Switch 开关语句	84
循环	87
While 循环语句	87
Do-While 循环语句	89
For 循环语句	90
其它语句	92
Break 中断语句	92
Continue 条件继续语句	93
Conditional 条件语句	93
Goto 语句	95
第 7 章 派生数据类型	97
符号表示	97
你会列举事物吗?	98
奇妙的枚举	98
类型定义别名 (typedef)	100
用数组组装数据	101
使用数组	103
多维数组	105
字符串及其它字符类型	109
字符串函数	110
串接字符串	114
比较字符串	114
用结构存储数据	115
结构数组	117
结构的嵌套	118
结构和类型定义	119
第 8 章 指针	120
扼要介绍	120
指针的创建	120
指针的初始化	121
指针的引用	122
动态内存分配	125
动态内存的分配	125
动态内存的释放	127
使用 C++ 的 new 和 delete 操作符	127
指针操作	128
指针和字符串	128
指针和地址操作符	131
指针和数组	133
指针和结构	136
第 9 章 函数	140
基本函数	140

调用函数	141
封装语句的函数	141
自动变量 (局部变量)	144
自动变量的实例	145
变量作用域	146
参数和变元	148
指针参数	151
数组参数	153
自顶向下编程方法	156
函数原型	157
头文件和原型	158
工程管理文件	161
第 19 章 高级编程技巧	163
再谈 main 函数	163
命令行变元 (arguments)	163
main 函数的 return 语句	164
退出和错误处理	165
详谈预处理指令	166
包含文件 (#include)	166
宏替换 (#define)	167
未定义符 (#undef)	168
条件编译 (#ifdef 和 #friends)	169
编译指令 (#pragma)	170
调试出错指令 (#error)	171
预定义符号	171
ASSERT.H 头文件	171
嵌套注释	172
再谈指针	173
指向指针的指针	173
指向函数的指针	174
再谈结构	177
不完整的结构说明	177
联合	178
位域	179
再谈变量	180
第三部分 标准函数库	183
第 11 章 输入输出函数 (STDIO.H)	185
STDIO.H 中的定义	185
文件处理函数	186
目录函数	187
文件 I/O 函数	189
字符 I/O 函数	191

字符 I/O 文件函数	193
格式化输出函数	195
格式化输入函数	199
错误处理函数	202
第 12 章 实用函数 (STDLIB. H)	204
STDLIB. H 中的定义	205
流程控制函数	205
ASC I 到二进制转换函数	207
二进制到 ASC I 转换函数	209
内存管理函数	211
环境函数	213
排序与交换函数	215
搜索函数	218
数学函数	220
随机数生成函数	223
第 13 章 时间和日期函数 (TIME. H)	225
TIME. H 中的定义	225
基本日期与时间函数	226
二进制转换函数	228
ASC I 转换函数	230
第 14 章 字符串函数 (STRING. H)	233
STRING. H 中的定义	233
实用函数	234
大小写函数	235
拷贝函数	236
比较函数	238
串接函数	240
匹配函数	242
其它函数	244
缓冲区操作函数	245
第 15 章 数学函数 (MATH. H)	249
MATH. H 中的定义	249
三角函数	250
对数函数	252
幂函数	253
其它函数	253
第四部分 算法	257
第 16 章 用户自定义数据结构	258
链表	258
单链表	260
链表的推入操作	261
链表的弹出操作	261

链表的遍历操作	262
链表的范例程序	262
树	265
树的前序遍历	266
树的中序遍历	266
树的后序遍历	267
树的范例程序	267
第 17 章 排序和搜索	272
排序	272
有关术语	272
选择排序	273
插入排序	275
搜索	277
第 18 章 文件操作	283
文件和头文件	284
文件格式和支持函数	285
文件的基本操作	305
文本文件	313
二进制文件	320
顺序存取方式	324
随机存取方式	326
目录	331
排序	335
结束语	341
附录 A 优先级和求值次序	342

第一部分 C 语言基础知识简介

如果你已决定学习 C 语言，那么在此你即将开始在程序设计领域里激动人心的历程，而这种难忘的经历也许会改变你今后的生活。当然，读者也许是一位就读于计算机科学专业的学生，或者可能打算在业余时间从事程序设计工作；读者也许已经掌握了另一门编程语言，而现在只想速成 C 语言；或者，读者仅想知道，在专业软件开发人员中，为何 C 语言成为众多编程语言中首选的编程语言。

无论读者希望学会用 C 语言编程的理由是什么，选择 C 语言都是很明智的。本书第 1 章介绍了 C 语言、C 语言的发展史、设计目的和它对程序设计者的价值。同时，作者还提出了学习编程的一种快速、易行的方法——“即学即用”方法。这一切仅需安装本书附带的 Broland 公司 Turbo C++ 编译器的专用版本，而不需要再购买其它任何软件就能实现。（当然，首先需要有一本书，它能解决如何组装一台 PC 机，然后才能安装计算机系统。）最后，为了激发对下一步操作练习的兴趣，读者可尝试运用两个“即学即用”的示例程序。

第 2 章讨论了 Turbo C++ 的一些操作。诸如编辑器、编译器、调试器及选项等。如果读者已具有一些 PC 机方面的经验，则可略过本章的内容。Turbo C++ 的集成开发环境 (integrated development environment, 简称 IDE) 提供一种“一步到位”的 C 语言编程系统。这系统包含用户所需的一切操作，如编写源代码、执行程序 and 调试本书中的示例程序等。书中所有范例的清单都存放在磁盘上。

具有了这些基础知识之后，你将在第 3 章中浏览 C 程序设计语言。在本章中，你将了解到 C 语言的每个重要特征。而学完第 3 章后，读者就能判断出 C 语言是否是读者应该选择的正确语言，是否满足自己的要求。如果继续学习第二部分的话，读者就能更深入地了解 C 语言世界。

准备好了吗？请翻到下一页，开始学习吧！

第1章 即学即用方法

贯穿本书的一个简单主题思想是：只要会打字，用户就能学会用 C 语言编程。

读者的打字速度无论是惊人的快，还是象龟兔赛跑那样的慢，这都没关系。即不必担心打字技能如何不好，读者只需将书中醒目的程序清单敲到磁盘上 Borland 公司的 Turbo C++ 专用版本中，按照“即学即用”这种方法，读者就能学会用 C 语言编程，而且在这种环境下学习 C 语言远比在其他环境下的学习进步得快。

作者正是出于这个浅显的道理，而大胆地作出上述声明的。不久前，作者在自己主持的《PC 技术》栏目的“Shades of Windows”一文中，曾抱怨“再没有人敲入杂志上的程序清单了”。随后，作者又提出一个要求——希望读者尽量自己敲入文中所列的程序清单，据此来告诉作者其观点是否正确。

令作者惊讶的是，读者们确实那么做了。于是，作者只得收回自己讲过的话（“食言”）。即使在数月之后，作者仍能收到读者的纷纷来信，十有八九的都说：即使盘上有程序清单，他们仍自己敲入所有源程序到编辑器里。当作者问他们为什么要这样做时，他们都说：“自己动手敲入源程序，能帮助我们了解程序是如何运行的。”

受这种想法的启发，作者提出了一种教学方法：只需将专用的程序清单敲入到计算机中，就可帮助读者毫不困难地学会 C 语言。这是一种快捷、易学的 C 语言编程方法；更重要的是，它有助于牢记所学的东西。以上就是“即学即用”方法行得通的依据。

敲过一次，记忆永久

也许人人都有如此的经历：在一家杂货店里，当你的手伸进口袋时，才发觉曾精心准备的购物清单遗忘在厨房的桌子上了。而你这时又没有时间回家去取，于是你只能凭记忆来完成这次采购任务了。令人惊讶的是，你往往能买到购物清单上几乎所有列出的物品。这又是怎么回事呢？

道理很简单：只需将事物记录下来就能明显地提高学习和记忆能力。

作者的“即学即用”方法正是基于上述道理的。在本书的每一章节中，读者阅读编程术语和过程——诸如函数是如何执行的；接着，就需读者将磁盘上所有程序清单中相应的部分装入到计算机中去；然后，读者可按照每章节的提示，即将书中用黑体字书写的语句也敲入到计算机中，才结束了整个练习过程。如果读者想了解程序的轮廓，可以仅键入程序清单中的某些部分。另外，作者还精心挑选了做过标记的每个小段和部分程序，以便帮助读者把注意力集中于正讨论的主题上。在掌握新的程序设计方法时，读者仅需键入现成的程序清单，这样就可以节省时间。除非读者确实需要用到程序的某些部分，否则，可以先不考虑那些部分的存在。

为何我们选用 C 语言？

编程语言的目的就是使得程序员的工作变得轻松简单。要不是 C 语言的出现，程序员就得象过去那样编写软件，即触发设定在计算机内存中的每个位 (bit) 开或关 (即状态“1”或“0”)。很明显，这在当前繁荣发达的软件开发领域中几乎是行不通的。(即使是回顾一下那时的情景，也不会令人有多少乐趣。) 如今，程序员们正用 C 这样的语言编写各种各样的计算机程序。

C 语言为程序员提供了三个重要的优点：

▶ C 语言是一种通用编程语言。用 C 语言可以编写出游戏、商用软件、实用软件、数学模型、文字处理软件、图表软件和其它的 C 语言软件。

▶ C 语言是一种结构化编程语言。它采用了 while 和 for 结构化循环语句，并用它们代替了极易误用的 goto 语句。如果用非结构化语言 (如 BASIC 和汇编语言等) 编程，goto 语句的使用可能使错误和故障潜伏在所编写的程序中。

▶ C 语言是一种标准化编程语言。通常，程序员所说的 C 语言是指公认的标准 C 语言——ANSI C 语言。本书提供的软件是 Turbo C++ 编译器的专用版本，它可适用于 ANSI C 或 C++ 环境下的编程。

学习 C 语言的最重要两点理由是 C 语言的通用性和使用的普遍性。任何专业程序员都应熟悉精通 C 语言；而且，如今不用 C 语言作为软件开发语言的程序设计工作室几乎不存在。数不胜数的教科书、参考书和软件包都包含了 C 语言的程序。而且现在的 C 语言也为读者学习其他的计算机语言提供了一个极好的训练机会。在掌握了 C 语言之后，读者就可以根据自己的意愿，毫不困难地挑选其它程序设计语言。

什么是编译器？

这个问题将稍后解答。现在首先考虑什么是计算机语言。人们不用这种语言交谈，而且它的词汇也不在字典里——或者，至少这些词汇在一个程序里有不同的含意。当然，读者也就不能用 C 语言给母亲写家信了。(如果你这么做了，那么你和你母亲常常连语言表达都很困难。)

既然 C 不是一种传统语言，那么它是什么呢？而且 C 的编译器究竟又是什么呢？首先最关键的一点是，称 C 为一种语言是用词不当的。C 实际上是一种符号指令代码，即能在计算机上执行的命令集。显示屏上显示文本、两数相加以及向磁盘转送数据等等，所有类似的和其他众多的操作均可用 C 来编程实现。当然，这需要遵守 C 的约定和规则；换句话说，就是要符合 C 语言的语法。下面就是用了一些 C 语句的例子：

```
int count = 0;
while (count < 100)
{
    printf("count == %d\n", count);
    count++;
}
```

现在请勿担心这些语句看上去似乎很难懂。到最后，这些语句就象周末小品剧一样易懂。然而，要注意的是，即使读者不能理解这些语句，但仍能指出有个整数（简化为 int），其名为 count，并看出当 count 值小于 100 时，要重复执行括弧中的语句。这对括弧封装的语句构成一个程序块，在本例中，它先打印变量 count 的值，然后将 count 值增大，最终将使 while 循环终止而退出。正如如上所述，即使读者对每处细节并不理解，但 C 的可读性仍是它主要的便利之处。因 C 编程时，程序员用的是自己的语言，而不是用计算机的语言。

C 编译器的功能在于读入用 C 语言编写的符号语句，然后将其翻译成最终的程序（参见图 1-1）。这个过程称为编译（compiling）或汇编（compilation）。程序的源文件是普通的 ASCII 文本文件，它包含了程序指令，当它经编译后，就可以生成可执行代码。如果编程者有编程技巧（幸运的话），程序就可以按预先设想的那样准确执行。经过编译后的程序，可以作为独立于诸如 DOS 或 Windows 这样的操作系统的应用程序来执行。

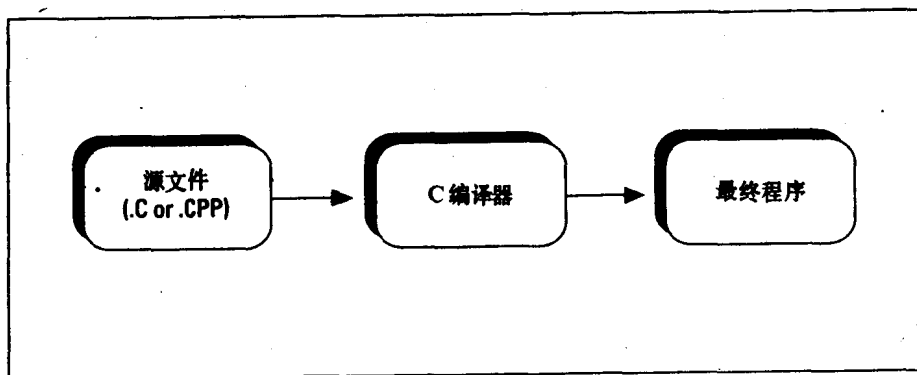


图 1-1 编译器将 C 语句翻译为最终程序

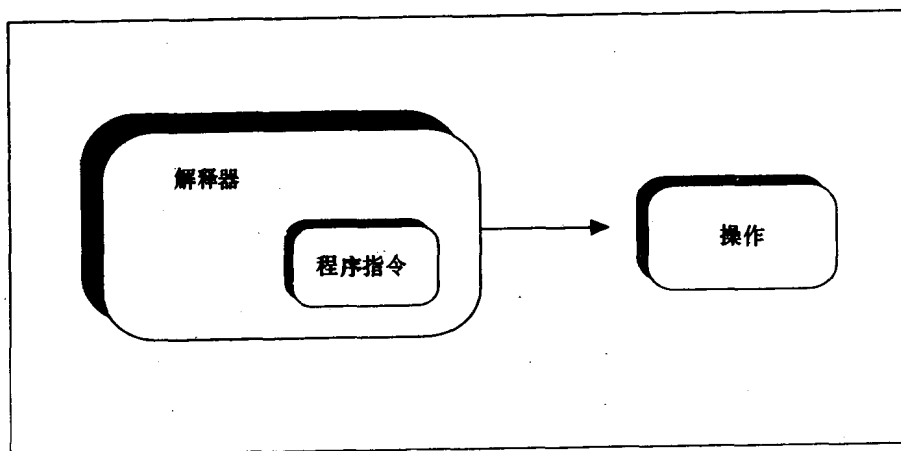


图 1-2 解释器在程序执行时，每次解释执行指令。

只有先运行了解释器后，才能执行程序。

并不是所有的编程语言都要被编译，例如，BASIC 的许多版本就使用解释器，而解释

器在每次程序执行时，才解释并执行程序语句（参见图 1-2）。由于 BASIC 是解释执行程序，解释器就需和程序一同运行。解释过程要花费很多时间；相反，由于编译后的代码是机器码形式，因此，编译型程序通常执行速度要快于解释型程序。另一方面，解释型指令一旦编写完，就可立即被执行；而用 C 编写的程序需要被编译，这就增加了程序开发所需时间。尽管如此，专业程序员都乐意采用象 C 那样的编译型语言，这是因为 C 能编写出人们能理解的高级语句，而又可产生能在计算机上快速执行的编译过的低级代码。

什么是连接器？

前面并没有说明所有的真实情况。编译器实际生成的是原始代码，即所谓目标代码，它们通常存贮于后缀为 .OBJ 的文件中。在目标文件送到 CPU 处理之前，必须先与其它部分相连接，生成后缀为 .EXE 的文件，即可执行代码文件。C 程序的编译过程仅完成了整个过程的一半任务。还需将目标代码连接起来，才能产生可执行代码文件（参见图 1-3）。

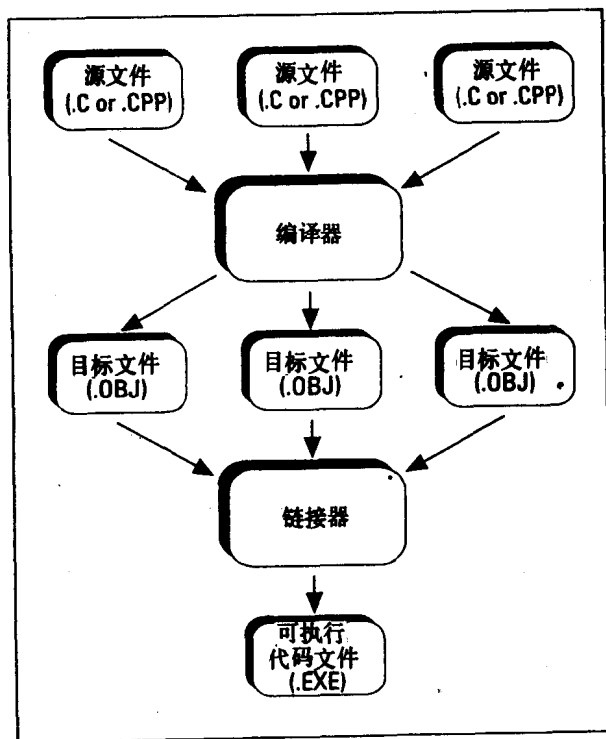


图 1-3 编译器将源文件翻译成目标代码；连接器则将目标代码文件组合成最终的可执行代码文件。

连接器也把操作系统所要求的启动和停机指令，与编译器提供的程序设计的其它部分连接到程序的目标代码中。连接器的首要任务是多个目标代码模块组成一个可执行代码文件。这要求程序员在编写大程序时，每次仅编写一小块程序，以保证最常需要的任务大小控制在可管理的范围内。