

Special Edition

C/C++

快速进阶教程

LEE ATKINSON 著
MARK ATKINSON



QUE

希望

科学出版社
龍門書局

**Complete Programming
Tutorial & Reference!**

For the Experienced Programmer
Switching to C or C++

Detailed Exercises for Procedural
C & Object-Oriented C++
Programming

Programming Tips & Techniques

Using C/C++ Special Edition

C/C++快速进阶教程

L. Atkinson M. Atkinson 著
陈晰 梁帆 译
梁帆 校

科学出版社
龙门书局

1995

(京)新登字 092 号

内 容 提 要

C/C++ 编程技术的魅力是无穷的。在计算机编程,尤其是 C/C++ 编程中,你可以用新的结构重建您的应用程序。本书介绍了一种非常适合于创建并修改计算机环境的语言。C 语言具有高级汇编语言的特性,很容易用它来控制机器的硬件。同时,它也是一种适合于编制高效、复杂程序的高级语言。本书适合于从事程序设计的各类人士参考。

需要本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮政编码 100080,电话 2562329。

版 权 声 明

Authoried translation from the English language edition published by Que Copyright © 1993.

Chinese language edition published by Beijing Hope Computer Company & Science Press and Longman's Book Co., Ltd./Simon & Schuster (Asia) Pte Ltd Copyright © 1995.

本书英文版名为《Using C/C++ Special Edition》,由 Que 公司出版,版权归 Que 公司所有。本书中文版由 Simon & Schuster (Asia) Pte Ltd 授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

C/C++ 快速进阶教程

L. Atkinson, M. Atkinso 著

陈 晰 梁 帆 译

梁 帆 校

责任编辑 魏玲

科 学 出 版 社
龙 门 书 局 出 版

北京东黄城根北街 16 号

邮政编码:100717

施园印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1995 年 6 月 第 一 版 开本:787×1092 1/16

1995 年 6 月 第一次印刷 印张:39 1/4

印数:1~5000 字数:919000

ISBN 7-03-004912-8/TP·477

定 价: 47.70 元

作者简介

Mark Atkinson

Mark Atkinson 1969 年开始从事计算机程序设计,当时参加编写“跟踪 Saturn-V 火箭和月球卫星”的 FORTRAN 程序。从那时起他就一直从事计算机研究,包括网络、编程和写作。他是 Que 公司出版的《Using Microsoft C/C++ 7》和《Using Borland C++ 3》两书的作者,现在他在美国法院任系统分析员。

Lee Atkinson

Lee Atkinson 有着 20 年从事数据处理工业的经验。他能够熟练应用 C、Pascal、COBOL、FORTRAN、PC/I、APL 和从微机直到 IBM 巨型机上的一系列汇编语言进行编程。现在,他是密西西比地区销售公司的 MVS 系统程序员,是 Que 公司出版的《Using Microsoft C/C++ 7》和《Using Borland C++ 3》两书的作者。

致 谢

我们衷心感谢下列同仁和组织：

—— Que 公司的编辑和生产人员，尤其是 Joe Wikert, Kezia Endsley, Ginny Noble, Chuck Hutchinson 和 Judy Brunetti, 感谢他(她)们的耐心指导。

——美国国家标准局, 感谢它允许作者引用 X3.159 的一部分。

目 录

绪论.....	1
---------	---

第一篇 C 语言介绍

第一章 C 语言入门	6
1.1 C 语言初探	6
1.2 C 程序的组成	12
1.3 C 程序编制	20
1.4 用 C 环境编程	22
1.5 小结	29
第二章 I/O 编程基础	30
2.1 输入输出的有关问题	30
2.2 数据流 I/O	32
2.3 简单的 I/O 功能简介	39
2.4 格式化 I/O 函数	55
2.5 硬件复制和系统独立性	61
2.6 小结	67
第三章 构造 C 程序	69
3.1 C 程序中的数据管理	69
3.2 函数说明及参数传递	89
3.3 小结	101
第四章 C 程序设计详述	102
4.1 语言规则的细节	102
4.2 使用编译命令和宏	116
4.3 小结	126
第五章 对象、表达式、操作符和转换	127
5.1 对象作用域和生存期详解	127
5.2 存取外部数据和函数	132
5.3 理解 C 的表达式	142
5.4 C 的操作符	146
5.5 类型转换	161
5.6 小结	164
第六章 C 的函数	165
6.1 参数传递	165

6.2	向 main() 传参数	174
6.3	递归函数	177
6.4	函数使用的补充说明	182
6.5	小结	194
第七章	程序逻辑流的控制方法	195
7.1	循环	195
7.2	条件逻辑编程	214
7.3	程序的提前结束	222
7.4	系统命令处理程序的调用	224
7.5	ANSI 标准的常见扩展	225
7.6	小结	225
第八章	用指针、数组和字符串编程	226
8.1	指针与复合数据类型	226
8.2	定义变量数组	235
8.3	用数组表示几何曲线的光滑连接	241
8.4	用数组来解方程组	246
8.5	字符数组与字符	249
8.6	充当函数参数的数组和字符串	262
8.7	小结	263
第九章	指针的使用	265
9.1	用指针表示指针	265
9.2	用指针扫描、分析文本	266
9.3	数组与指针的混合使用	285
9.4	用指针提高程序性能	287
9.5	小结	294
第十章	文件输入/输出(I/O)编程	295
10.1	C 的文件管理函数	295
10.2	缓存 I/O 的概念	299
10.3	文件流的 I/O 模式	302
10.4	直接存取文件编程	308
10.5	用杂凑关键字直接存取	313
10.6	编写高质量的文件例程	320
10.7	对 ANSI 标准的公共扩展	323
10.8	小结	324
第十一章	派生出新的数据类型	325
11.1	结构的定义	325
11.2	联合	338
11.3	用 typedef 派生新的类型	340
11.4	数据的动态管理	341

11.5	作为函数参数来使用的结构和联合	345
11.6	用结构建立链表	345
11.7	枚举常量的使用	364
11.8	小结	365
第十二章	C语言移植及转换要点	366
12.1	保留C语言的精华	366
12.2	销售商和程序员之间的协议	367
12.3	未指定的操作	367
12.4	未定义的操作	368
12.5	定义实现的操作	372
12.6	C语言的可移植性和兼容性	375
12.7	环境考虑	378
12.8	小结	381
第十三章	如何从其他高级语言转向C	382
13.1	BASIC向C的转换	382
13.2	从Pascal向C转换	396
13.3	小结	408

第二篇 C++编程基础

第十四章	对象及面向对象的编程	412
14.1	对象是工作模块	412
14.2	类是抽象数据类型	418
14.3	类的继承性是类型派生	424
14.4	面向对象系统的定义	424
14.5	面向对象系统的其他要点	427
14.6	小结	428
第十五章	定义类和对象	429
15.1	定义C++的类	429
15.2	类对象的初始化及释放	439
15.3	类对象的使用	444
15.4	小结	487
第十六章	类与对象的控制	488
16.1	C++的自由存储	488
16.2	定义C++的对象	500
16.3	派生类和继承性	504
16.4	C++数据流I/O操作	518
16.5	小结	521

第十七章 C++方法及对象详述	522
17.1 C++中的指针和引用	522
17.2 重载函数.....	530
17.3 重载操作符.....	532
17.4 C++扩展	550
17.5 小结.....	556
附录 A ASCII 字符集	557
附录 B ANSI 预定义宏	564
附录 C ANSI C 库函数	569
附录 D 程序 FINANCE.C	576
附录 E IBM PC 通信程序设计	591
附录 F B-条样曲线推导	610
附录 G 性能评价软件	611

绪 论

C/C++编程技术的魅力是无穷的。在计算机编程,尤其是C/C++编程中,你可以用新的结构重建您的应用程序。

本书向您介绍了一种非常适合于创建并修改计算机环境的语言。C语言具有高级汇编语言的特性,很容易用它来控制机器的硬件。同时,它也是一种适于编制快速、复杂程序的高级语言。

由于C语言对这两种特性兼而有之,因此近年来它特别流行。虽然C语言并不是万能的,但它功能非常丰富,从运算、科学应用到编制操作系统,它几乎都可以胜任。

本书包含了C/C++编程设计时经常用到的基本技能。本书的宗旨在于帮助你编出上乘的应用程序。但是,本书并不是一本百科全书。因此,本书的范围受到以下限制:

- 虽然本书的篇幅受到限制,但仍然保证了所述内容的深度。目录表已经很清楚地显示出现代C/C++的基本要素,本书有效地覆盖了这些内容。因此,学习了本书内容之后,你的编程技术就能跨上一个新的台阶。
- 学习语言时用的是简单程序,而不是特定的应用程序。介绍大型的复杂程序并不是本书的目的,本书的目的在于向你介绍构成大型程序的手段。
- 本书介绍C的标准内容。在K&R C之后是ANSI C和C++(ANSI C再加上面向对象的扩展部分)。本书的重点在于ANSI C,由此你可以掌握所有C编程的必要概念和技巧。只有在你掌握了C的这些概念之后,才能很顺利地过渡到C++的学习。

本书的目的

本书的目的在于向你介绍C/C++编程,由此你可以从初学者过渡到中等程度的程序员。如果你对C/C++编程不熟悉,可以把本书当作工具书来用。本书有以下三个目的:

- 本书内容是现代C/C++编程的基础,它包括ANSI C和C++编程。因此,你不必再去买好几本书一起看。本书为你提供了开始C/C++编程的全部知识。
- 本书有效地向你教授C/C++语言。如果方法不符,C/C++编程的学习会变得很困难,而本书就帮助你去掉了这种担心。我们为你安排了C/C++学习的最有效的途径。
- 本书可以作为一本极有价值的编程参考手册来使用。为此,我们在书中用图文并茂的形式讲述,并提供了参考和小结等。因此,即使你学完了本书内容,也还可以把本书作为快速查阅时的必备手册。

本书适合的读者

本书特别适合于初级或中级的C/C++程序员。你可以通过本书完成程序设计的入门,也可以用此提高编程的水平和技巧。但是本书对那些对C/C++编程有一定经验的人也相当有价值。如果你对C/C++的传统编程方法很熟悉,本书有助于你迅速适应现代的编程尤

其是小结和参考内容部分,它们对所有 C/C++ 的程序员都极有参考价值。

本书并不是针对具体的编译器产品而写的,它介绍的是标准 C/C++。但是,本书中也提及了一些特定的编译器环境,这样做只是为了强调例子能实际运行的环境,所以,范例程序都标明了所用编译器的种类本书用到下列两种编译器:

Borland C++ 3.1 (Borland)

Microsoft C/C++ 7.0 (Microsoft)

同时,本书还提及了不同编译器对 ANSI C 的扩展功能以及它们彼此之间在某些功能实现上的差异。

本书所需的预备知识

要学习 C/C++ 编程的内容,必须先具备常用的计算机操作知识以及普通的编程概念。如果在学习本书内容之前你并未接触过计算机编程,那么学习过程中编程的练习量应相对来说要大一点。C 语言的表现形式灵活多样,对于那些疏于使用计算机的人来说是不容易掌握的。因此,在学习 C/C++ 编程之前你应该先熟悉计算机的使用。

如果你对计算机完全陌生,那么不妨先看一些基本的计算机资料。

本书的编排

本书分为两部分,一部分是 ANSI C 基础,另一部分是 C++ 基础。第一部分介绍 C 语言的构成机制,第二部分则介绍 C++ 的学习。

符号与约定

在本书中,新的和需要强调的内容都用斜体字 (*italic*) 表示。函数、命名、参数都用黑体字 (**bold**) 表示,比如, `main()` 函数就是这样。注意,在下列语句中的黑体部分:

```
Please Enter Your Name:Erin
```

代码行中的占位符号(在实际编程中应使用所需要的字符串来替换它们)用单斜体表示。比如:

```
long double variable1, variable2;
```

本例中, `variable1` 和 `variable2` 可用适当的字符串来取代。

一个完整的 C/C++ 程序,在清单的开始部分都有一段头代码。在整个程序中,每行代码都标注了行号。这样,要说明某一特定行就比较容易。只有在程序很短时才不写行号和头代码。本书还有一些内容框表示警告、注意及参考等意义,例如:

警告:警告框是用来提醒你需注意的问题的,包括程序中可能引入的出错部分,或是可以引起系统崩溃的部分。

注意:注意框向你提供一些特别的信息。有时这些信息有助于加速学习 C,并向你提供一种 C 编程的捷径。有时,只是提醒你注意一些很重要的内容。

小结与参考框

参考框和小结框都是注意框,框中的标题有以下四种情况。

- **ANSI C 理论。**这种注意框包括一些 ANSI 标准的提法。这有助于对 C 语言的完整理解。
- **传统 C 编程方式。**在 ANSI 标准公布之前已经有和当数量的 C 应用程序。这些程序框有助于读懂传统语法的 C 程序,并有助于这些程序向现代语法的转换。
- **编译器的依赖性。**由于 C 可心在各类的机器和操作系统上运行,而各种机要都提供了一些各自的功能。因此,如果使用不止一种编译器,你就有必要知道一些这种编译器的软硬件环境的特点。
- **快速参考。**一旦需要对 C/C++ 语言做某种小结时,就会出现这种注意框。这种注意框有助于快速复习一些与书中内容相关的知识。

本书的使用方法

本书既可以作为 C/C++ 学习教材,也可以作为该语言的参考书。

本书是面对 C/C++ 编程的。按照书中内容的排列顺序,你可以领略 C/C++ 程序设计的全貌。在书中每个范例程序后面都给出了详细解释——这样你可以在实践中领略 C 的奥妙。而不必去死记硬背 C 语言的规则,因为通过大量的编程实践你自然就能灵活动运用这些规律了。

本书也可以作为一本参考书来使用。书中的表格内容很详细,你可以通过目录来查找内容的位置,再通过相关的 C 注意框和表格来查阅所需要的内容。



第一篇 C 语言介绍

- 第一章 C 语言入门
- 第二章 I/O 编程基础
- 第三章 构造 C 程序
- 第四章 C 程序设计详述
- 第五章 对象、表达式、操作符和转换
- 第六章 C 的函数
- 第七章 程序逻辑流的控制方法
- 第八章 用指针、数组和字符串编程
- 第九章 指针的使用
- 第十章 文件输入/输出(I/O)编程
- 第十一章 派生出新的数据类型
- 第十二章 C 语言移植及转换要点
- 第十三章 如何从其他高级语言转向 C

第一章 C 语言入门

如果您想学习 C 语言的基本原理,那么您可以在本书中找到与之有关的所有内容;如果您刚开始接触 C 编程,本章将给您介绍入门的基本知识;如果您对 C 已有了一些经验,那么本章将帮您对此做个回顾。

1.1 C 语言初探

学习新语言时,除了要了解其正确的数据类型,还要对这门语言的概貌有个大致了解,用图表来显示背景将使对学科的了解更加具体化。本章将向您介绍 C 语言的概貌,主要包括如下内容:

- C 的历史,从 K&R 到 C++
- C 与其他语言的比较
- 如何读懂一个 C 程序
- C 程序的基本结构
- 编写 C 程序的步骤

1.1.1 C 程序的历史简述

C 语言来自何方?它是如何发展的?它有哪些优点?这些问题的答案虽然无助于 C 语言的学习,但有助于您对 C 语言的背景、连续性及与 C 相关的事情有一个大致的了解。

1.1.1.1 C 语言的诞生

1978 年,Prentice Hall 公布了由 Brian. W. Kernighan 和 Dennis. M. Ritchie 合作研制的 C 编程语言,这就开始了对 C 的探索。

最初,由于 Dennis. Ritchie 在 UNIX 操作系统中的使用而产生了 C 语言,这时它用在新泽西州墨累河贝尔实验室的 DEC PDP- I 计算机上。当时除了实际的编译器外,UNIX 操作系统及其大多数应用程序都是用 C 写的。

除了 Kernighan 和 Ritchie 二人,很少有人能解决这个问题,UNIX 的 C 源于 Martin. Richards 的 BCPL 语言,这种语言强烈影响了 B 语言的发展。由 Ken. Thompson 研制的 B 语言在 1970 年用于 DEC PDP-7 上最初的 UNIX。1972 年,Dennis. Ritchie(当时他与 Ken. Thompson 共事)把 B 语言扩展为 C 语言,它比 B 语言多了一种东西:数据类型。

BCPL 和 B 都被设计成非类型语言,但 C 却有三种基本类型:整数、字符和浮点数。有这三种数据类型,就可以很容易地控制程序操作的数据。

一段时间里,C 语言一度很神秘、很难学。但由于 Kernighan 和 Ritchie 二人公布了 C 的秘密,C 很快被推广到 IBM 等公司。现在,C 语言几乎已经路人皆知了。

1.1.1.2 C 语言进入 DOS

C 从 UNIX 到 DOS 的过渡并不是一帆风顺的,最初 C 一度用于 8 位微机的 CP/M 操作系统。虽然 CP/M(它由数字研究所从 UNIX 降低规模而得到)使用也很普遍,但这段时间 C 并没有真正得到发展。其原因是显而易见的:8 位机系统一般只有 64K 的 RAM 容量,这对 C 而言是极大的制约。

1980 年,16 位微机系列 8086/8088 产生了,这打破了微机存储容量的限制,C 语言也开始得到了发展,同时(也在 1980 年)西雅图计算机制造公司推出 86-DOS,DOS 操作系统开始设计和销售。从此,C 的使用很快从 CP/M 上转移过来。

Microsoft 和 IBM 两大公司的出发点是保证软件公司将为新机型和操作系统生产系列产品,包括编译程序。IBM PC 及其配套软件的成功应部分归功于那个环境下的 C 编译程序的普及,C 把“象牙塔”中的程序员变成喜欢编码的群体。

1.1.1.3 C 语言引起 ANSI 的注意

一些早期的 C 作者认为因为仅有两个人(Ken. Thompson 和 Dennis. Ritchie)的参与,C 语言是很“纯正”的,他们蔑视委员会为了编译程序标准化而做的努力。

而事实是,ANSI(美国国家标准协会)已经着手定义其他语言的标准,例如 COBOL。这种标准化的最大的好处是“可移植”,即从一个标准的 ANSI 编译器中把 COBOL 应用程序移植到另一个编译器。这种移植说明一个问题,完全可移植性是不可能实现的,即使是同一个制造商生产的两个编译器,也会出现些微的差异。

虽然 ANSI 对 C 很感兴趣,但在 C 语言并未唤起程序员们的热情时,要想让他们对 C 满意也是不现实的。而且,由此就对 ANSI 的标准化进行讽刺也是不公平的。为了完成 C 的标准定义,ANSI 对 C 花了比对 COBOL 更多的努力。

1.1.1.4 C 语言的标准化的

ANSI 建立了 X3J11 委员会,目的就是组合“标准的 C 语言”。委员会面临的主要问题是规范现存的各种语法形式,换言之,不论何时,新的标准都能兼容过去的语法形式,委员会在这一点上取得了成功。C 语言保持了 K&R 两人所作的定义,但去掉了语言中的漏洞和不一致性。

这条规则说明:多数已有的程序编码可以运行在新标准的编译环境中,C 语言的编码由此丰富起来,而且带来了巨大商业价值,这种情况直到标准不被市场接受时才会消失。

1.1.1.5 ANSI C 的确立

并无条文规定编译程序制造商必须用 ANSI 标准来规范自己的产品。但是,销售时尽管自己的产品并不与 ANSI 兼容却硬要这么说,这是不明智的。对用户的欺骗会直接影响产品的销量,实际使用的程序常常必须使其宿主软件和操作系统不仅要达到一定的功能级别,还要支持一定的函数。

ANSI 和编译器销售商都认识到对标准作性能上的某些扩展是完全必要的。但特别的技巧不应作为扩展标准的一部分。这样,您会发现 C 编译器一般都提供了 DOS 的界面,可以存取 I/O 接口,控制显示等等。事实上,如果希望程序功能完备的话,编译器的性能扩展是非

常必要的。

1.1.1.6 C语言的可移植性

可移植性是一种无需修改就可在多个销售商的编译器上运行自己程序的能力。这就要求标准应尽可能地有影响力,不能支持可移植性的标准毫无用处。这也是像C这样的语言影响面如此之大的原因所在。C的一个很大的好处就在于,该语言可以直接控制大部分系统环境而无需汇编语言的介入,尽管各种环境各不相同。当然,100%可移植的编译器是不存在的。ANSI委员会决定ANSI标准应给程序员以编制可移植代码的选择,而不能强迫他们这样做。这对程序员来说事情就有点复杂了。对转换和可移植性的完整讨论将在第十二章介绍。

1.1.1.7 正在被看好的C++

ANSI标准并未使C语言流行起来,相反,是因为C的流行才促成了它的标准化。从1988年十二月开始,ANSI X3J11开始关心C++的标准化问题了。当然,C++的标准化工作还在继续,但是C++不再是像C那样作为研究实验的秘密工具了。C++日益成为程序员有力和可移植性强的编程工具。

注意: Borland 和 Microsoft 公司的新 C/C++ 软件包强化了编程手段,如果您买了这些厂家的 C++ 软件包,那么您的编译器一定是功能齐备的,尽管您还不想马上用 C++ 语言来编译,但由于 Borland C++ 或 Microsoft C++ 具有优越的性能,采用这些编译器只是迟早的事。

1.1.1.8 面向对象的程序设计

面向对象编译系统,或称 OOPS(这是其缩写),已经流传有一段时间了。由于C的灵活性及功能特性,决定了有时性能缺乏。C++就保证这方面性能——它既是面向对象的语言又性能卓越。

C++不仅提高了程序运行速度,同时还提高了程序员的编程速度。C++使得您可以快速开发大量项目而不会多犯错误,同时提供了一些C语言所没有的工具。

1.1.1.9 贝尔实验室再立新功

C++是由Bjarne Stroustrup在贝尔实验室工作设计出来的,本书中引用的C++内容源于《C++编程语言》(此书于1986年由Addison-Wesley发表,目前已出了第二版是其再版文章)。

C++并不是完全从ANSI C中划分出来的,事实上,ANSI C可以看作是C++的子集,因为两种语言在方法和设计上都互相借鉴。C++与标准C距离很有限。C++支持一个很特别的存储方法:封装,数据隐含,类和派生类,函数和操作符重载等等。

1.1.2 C语言与其他语言的比较

和其他语言一样,C既有长处也有不足。这是因为C的体系提供强劲有力的功能来适应程序快速运行的需要,还提供了一个函数库和丰富的操作符集。因此,C比别的语言都要略胜一筹。