



软件工程师 宝典系列

C/C++ 程序设计

陈 策 张素卿 管佩森 等编著

- 提纲挈领，萃取C/C++语言语法要旨
 - 知识、实例、练习行文，实用、实效
 - 语言间横向对比，拓展编程技能
 - 轻松实战操作演练，培植个性能力
 - 编译、连接、调试深度剖析，简明、经典
- 近**150**个实例精解剖析，让你轻松步入顶尖软件工程师殿堂



科学出版社
www.sciencep.com

软件工程师
宝典系列

C/C++ 程序设计

陈 策 张素卿 管佩森 等编著

 科学出版社
www.sciencep.com

内 容 简 介

这是《软件工程师宝典》系列图书之一,本书是学习 C/C++ 语言的实用参考工具书,比较全面地讲解了 C/C++ 的基础知识及实例应用。

本书共分 11 章,第 1 章到第 6 章主要讲解 C 语言程序设计的内容,包括 C 语言的发展、C 程序的特点与基本构成、C 语言的数据类型、运算符与表达式、程序控制语句、函数以及预编译指令等。从第 7 章开始,主要讲解 C++ 语言在面向对象方面扩充的内容和特性,包括面向对象方法学的基本理论、C++ 程序的基本构成、类与对象、函数与运算符重载、继承与多态等。第 11 章专门对 C/C++ 中的输入与输出功能进行了讲解及对比。

各章都结合大量实例,分别对各个重要知识点进行实验。每部分都提供了上机操作和习题,有详细的开发实例,章后附录若干习题,书后附有习题参考答案,便于初学者学习、实践。本书内容全面、实例丰富、叙述清晰、结构安排合理,C/C++ 二者并用,其实用性、可操作性强,能够有效地提升读者的开发技能和经验。

书中有些实例来自实际项目,读者可以参考使用。本书适合有一定编程经验的 C/C++ 初学者,以及想要在 C/C++ 编程经验上得到快速提高的编程人员,同时也是社会培训班选择的理想教材。

本书部分实例源代码可免费从 www.bhp.com.cn 下载。

图书在版编目(CIP)数据

C/C++ 程序设计 / 陈策 张素卿 管佩森等编著. —北京:
科学出版社, 2008
(软件工程师宝典系列)
ISBN 978-7-03-022984-7

I. C/C++... II. ①陈... ②张... ③管... III. C/C++ 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 140823 号

责任编辑: 但明天 / 责任校对: 周 玉
责任印刷: 媛 明 / 封面设计: 久久度设计

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2008 年 11 月第 一 版 开本: 787×1092 1/16
2008 年 11 月第一次印刷 印张: 22 3/4
印数: 1—3000 字数: 524 628

定价: 35.00 元

总 序

计算机硬件技术的发展日新月异，CPU 几乎是按照摩尔定律进行快速更新的，随着时间的推移，其运算速度呈级数级地增长。同样，计算机软件技术也在突飞猛进地发展，无论是操作系统还是应用软件。

操作系统的开发是从第一代软件工程师在计算机硬件基础上进行的最低层的二进制编码开始的，到后来逐步发展到用户不需要熟悉低层计算机指令即可进行操作的 DOS 系统，再发展到所见即所得的 Windows 桌面操作环境，以及功能强大的 Linux、UNIX 系统。经过多年的发展，操作系统的功能及作用已经发生了根本性的变化。

同样，运行在操作系统之上的各种应用软件也在发生着根本性的变化，从以前需要对硬件有深刻了解才能编好程序的汇编语言开始，到后来广泛使用的高级程序语言 Basic、C、Pascal、FORTRAN 等，再到现在被广泛使用的 Visual Basic、Visual C++、Delphi、Java 语言及各种 Web 语言等，软件的操作越来越方便，功能却越来越强大，从而使软件工程师的编程变得越来越简单，不过要了解的知识也空前庞大。

随着信息社会的到来及对无纸办公需求的增加，现代社会对软件工程师的需求大量增加，仅我国软件工程师的缺口就在数十万以上。软件人才的培养决定了信息社会实现的程度及社会发展的速度，所以适时、合理地大力培养一些优秀的软件开发人才，对于我国信息化产业的发展必将起到举足轻重的作用。

正是基于这样的计算机软件发展背景及信息产业化发展需求的考虑，我们经过精心策划、周密设计，组织最优秀的一些作者，编著出版了一套常用计算机软件及操作系统的系列图书，希望使之成为软件人才培养及推动信息化产业发展的“宝典”。这套图书起点低，使读者入门快。同时每本书的内容都很实用，案例丰富简练，与基础知识一一对应。图书按照软件人才培养的规律，尽量使内容讲解由易到难、深入浅出。

该套丛书一改过去基础类图书中重说教、轻操作、过于注重理论及概念讲解的弊端，以一种全新的边操作、边熟悉、边学习的方式吸引读者深入学习下去。每本书在精心挑选并巧妙设计大量案例的基础上，将基础知识的讲解融合到案例的练习中，二者相辅相成，结合紧密。

通过案例的操作与分析，可以加深对基础性概念的理解，并以大量的上机操作来巩固所需要掌握的基本知识点。在内容安排上步步为营、循序渐进，在将基础知识学扎实、学透彻的前提下，使读者的软件编程技术也能够迅速得到提高，使他们的技术水平能够尽快地更上一层楼。

本系列图书面向的读者群体非常广，而对读者的要求又非常低，他们只要对计算机的基本操作熟悉，就能够学好本系列图书中的每一本，从而可以快速掌握对应的每一种软件。读者可以是大学、中专院校的在校学生，各种计算机培训班的学员，社会各个机构中需要培

训的在岗人员，以及所有对软件技术有兴趣的计算机爱好者，本系列图书都将成为他们计算机软件学习道路上最好的良师益友，最得心应手的“宝典”，最能够提高他们的学习效率及促进技术进步的法宝。

我们非常希望本系列图书及时出版，能够为所有使用该系列图书的读者奉献上一套精美丰盛的“计算机软件学习”满汉全席。

从书编委会

前 言

自 C 语言产生以来,一直受到计算机领域的研究人员、教育工作者和程序设计者的广泛关注。C 语言兼具低级语言和高级语言的特性,非常适合于编写系统程序。由它开发出的程序具有代码结构简洁、执行效率高、可移植性好、维护和扩展方便等特点。因此,C 语言曾一度被美誉为真正的程序设计者的语言。

面向对象方法是目前系统分析和程序设计领域最有效、最实用、也是最流行的一项技术。C++作为 C 语言的一个超集,将 C 扩展成为一门面向对象的程序设计语言,这进一步使 C 和 C++语言成为目前占主导地位的编程开发语言。C/C++中的许多概念和元素都是任何一门程序设计语言所必备的。因此,学好 C/C++,不仅可以为大家进入程序员的行列打下良好的基础,而且可以为大家迅速掌握其他编程语言作好准备。

本书从 C 语言的发展历史入手,系统、全面地介绍了 C/C++的语法元素、编程规则和使用技巧。另外,在介绍 C/C++语言的同时,还讲解了与它们有关的结构化分析、程序设计以及面向对象的分析与程序设计理论。

相对其他高级程序设计语言而言,C++在某些概念上更为规范,也更容易出错,例如指针、动态内存分配、流等。为此,书中列举了大量的实例以帮助读者消化和吸收所学到的内容。另外,在每一章的上机操作部分,也给出了一些针对具体应用背景的程序设计实例,所有这些实例都力求典型、易懂、实用。

要编写 C/C++程序,一定离不开相应的编程开发环境。目前,有许多用于 C/C++语言的集成开发环境,Visual C++就是其中之一。虽然 Visual C++集成环境非常复杂,但我们还是选用它作为 C/C++编程开发的练习环境,以便为大家今后的进一步学习和提高打下基础。

本书主要由陈策主编并通稿,姜新颜女士、王中许先生也参与了部分章节的编写并在写作中给予很大帮助,再此表示诚挚地感谢。另外,在本书编排过程中,张素卿、管佩森、黄丽娜、周鸣扬、李东玉、周松建、唐兵、刘旭、范荣、杜江、张俊岭、张江涛、冉林仓、刘伟、梁斌、张海霞、于丙超、范翠丽、薛年喜、杨小勇、刘秋红等也参与了部分内容的编写,在此一并表示感谢。

由于编者水平有限,书中难免有不当之处,敬请广大读者批评、指正。

编 者

目 录

第 1 章 C 语言及程序设计初步	1
1.1 程序设计语言的发展	1
1.2 C 语言的产生	3
1.3 C 语言的特点	4
1.4 C 程序的基本构成	6
1.5 C 程序的编辑、编译、连接和执行	8
1.6 C 集成开发环境 (IDE)	10
1.7 上机操作	16
1.7.1 建立第一个 C 应用	16
1.7.2 在 IDE 中排错、调试程序	18
1.8 小结	20
1.9 习题	20
第 2 章 数据类型、运算符和表达式	23
2.1 关于注释	23
2.2 标识符	24
2.3 数据类型	25
2.4 常量与变量	27
2.4.1 常量	27
2.4.2 变量	30
2.5 运算符	37
2.5.1 算数运算符	37
2.5.2 关系和逻辑运算符	39
2.5.3 位运算符	40
2.5.4 其他运算符	41
2.5.5 运算符的优先级	43
2.6 表达式	45
2.7 上机操作	47
2.7.1 字符常量的 ASCII 码	47
2.7.2 使用表达式	48
2.7.3 位运算符的应用	48
2.8 小结	51
2.9 习题	51

第3章 结构化程序设计	54
3.1 C语言语句	54
3.2 条件判断	55
3.2.1 if语句	55
3.2.2 switch语句	61
3.3 循环与循环跳转	63
3.3.1 while语句	63
3.3.2 do...while语句	65
3.3.3 for语句	68
3.3.4 break与continue语句	70
3.4 强制跳转	72
3.5 上机操作	76
3.5.1 顺序程序实验	76
3.5.2 分支程序实验	77
3.5.3 循环程序实验	78
3.5.4 综合实验	79
3.6 小结	80
3.7 习题	80
第4章 函数与编译预处理指令	83
4.1 函数的声明与定义	83
4.1.1 函数声明	84
4.1.2 函数定义	85
4.1.3 从函数返回	86
4.2 函数调用	89
4.2.1 函数调用的参数传递	89
4.2.2 函数的调用过程	91
4.2.3 数组作为函数参数	92
4.2.4 函数的嵌套调用	94
4.2.5 函数的递归调用	95
4.3 函数的作用域	98
4.4 主函数的参数	100
4.5 编译预处理指令	102
4.5.1 #include指令	102
4.5.2 #define与#undef指令	102
4.5.3 条件编译指令	104
4.6 上机操作	107
4.6.1 求整数的阶乘	107

4.6.2	使用递归函数.....	108
4.6.3	求方程的近似根.....	108
4.7	小结.....	110
4.8	习题.....	110
第5章	数组与指针.....	113
5.1	数组.....	113
5.1.1	一维数组.....	114
5.1.2	多维数组.....	115
5.1.3	数组的初始化.....	118
5.2	指针.....	119
5.2.1	指针与指针变量.....	120
5.2.2	指针运算符.....	121
5.2.3	指针的运算.....	123
5.2.4	动态内存分配.....	126
5.2.5	指向指针的指针.....	127
5.3	指针与数组.....	130
5.4	函数与指针.....	132
5.4.1	函数返回指针.....	132
5.4.2	函数型指针.....	133
5.5	上机操作.....	135
5.5.1	数组元素的冒泡法排序.....	135
5.5.2	使数组循环移位.....	136
5.5.3	动态内存分配与多重指针.....	137
5.5.4	字符串数组选择法排序.....	141
5.6	小结.....	143
5.7	习题.....	143
第6章	结构、联合及枚举.....	146
6.1	结构.....	147
6.1.1	结构与结构变量定义.....	147
6.1.2	结构成员的访问与结构变量的初始化.....	149
6.1.3	结构数组.....	150
6.1.4	结构与函数.....	151
6.1.5	嵌套结构.....	153
6.2	结构指针与链表.....	154
6.2.1	结构指针.....	154
6.2.2	链表.....	156
6.3	位域与联合.....	161

6.3.1	位域	161
6.3.2	联合	163
6.4	枚举	165
6.5	用户自定义类型	166
6.6	上机操作	168
6.6.1	使用结构数组	168
6.6.2	使用结构中的联合	170
6.6.3	利用链表实现信息管理	172
6.6.4	利用栈实现数制转换	175
6.7	小结	178
6.8	习题	178
第 7 章	从 C 到 C++	181
7.1	从结构化到面向对象	182
7.1.1	结构化程序设计的缺陷和不足	182
7.1.2	面向对象方法概述	183
7.1.3	面向对象的若干概念	185
7.1.4	面向对象的特征	190
7.2	从 C 到 C++	195
7.2.1	C++的发展历史	195
7.2.2	C++对 C 的扩充	196
7.2.3	C++程序的基本构成	200
7.2.4	C++程序的编译、连接和执行	200
7.3	C++函数扩展	202
7.3.1	默认函数参数	202
7.3.2	不定函数参数	204
7.3.3	引用函数参数及函数返回值	206
7.3.4	内联函数	208
7.3.5	函数的重载	209
7.4	new 和 delete 运算符	211
7.5	上机操作	213
7.5.1	建立第一个 C++应用程序	213
7.5.2	使用 C++的显式类型转换	214
7.6	小结	215
7.7	习题	216
第 8 章	类与对象	218
8.1	C++类与对象基础	219
8.1.1	类的定义	219

8.1.2	对象的定义	220
8.1.3	类成员变量和成员函数的定义	222
8.1.4	静态类成员	224
8.1.5	常量类成员	226
8.1.6	this 指针	228
8.2	对象的构造与析构	229
8.2.1	类的构造函数	229
8.2.2	类的析构函数	232
8.2.3	对象的构造及析构过程	233
8.3	嵌套类与局部类	235
8.4	友元	238
8.4.1	友元类	238
8.4.2	友元函数	239
8.5	运算符重载	240
8.5.1	运算符重载的形式与规则	240
8.5.2	使用成员函数重载运算符	241
8.5.3	使用友元函数重载运算符	243
8.6	类、结构、联合的关系	244
8.6.1	结构和类	244
8.6.2	类与联合	244
8.7	上机操作	245
8.7.1	使用类实现简单链表	245
8.7.2	为字符串类重载运算符	249
8.8	小结	251
8.9	习题	251
第9章	继承性	254
9.1	单继承与多继承	255
9.1.1	单继承	255
9.1.2	多重继承	257
9.2	子类对象的构造与析构	259
9.2.1	子类对象的生成与撤销过程	260
9.2.2	子类到父类间构造函数参数传递	261
9.3	继承中的成员覆盖与赋值兼容	265
9.3.1	继承中的成员覆盖	265
9.3.2	继承中的赋值兼容规则	266
9.4	多重继承的二义性问题	267
9.5	上机操作	270

9.5.1	使用类的单继承	270
9.5.2	使用类的多重继承	273
9.6	小结	276
9.7	习题	277
第 10 章	多态性	280
10.1	虚函数	281
10.1.1	虚函数的多态性	281
10.1.2	使用虚函数的意义	284
10.1.3	破坏虚函数动态链接的情况	286
10.1.4	虚析构造函数	287
10.2	纯虚函数与抽象类	289
10.3	虚基类	291
10.3.1	引入虚基类的目的	291
10.3.2	虚基类与虚函数结合时的多态性	294
10.4	使用运行时多态性	295
10.5	小结	300
10.6	习题	300
第 11 章	输入、输出和磁盘文件	304
11.1	文件和流的概念	305
11.1.1	文件的概念	305
11.1.2	流的概念	306
11.1.3	文件与流的缓冲处理	306
11.2	C 中的简单控制台 I/O	307
11.2.1	显示器输出函数	307
11.2.2	键盘输入函数	308
11.2.3	C 中的格式化控制台 I/O	309
11.3	C 中的 I/O 文件系统	313
11.3.1	文件打开与关闭	313
11.3.2	文件的读写操作	314
11.3.3	文件位置指针维护函数	315
11.4	C++ 中 I/O 介绍	316
11.5	C++ 的流输入与流输出	318
11.5.1	插入运算符与提取运算符	318
11.5.2	重载 << 与 >>	319
11.6	C++ 中流的格式化	321
11.6.1	格式控制标识	321
11.6.2	格式标识设置函数	322

11.6.3	其他格式设置函数	323
11.6.4	格式操纵符	324
11.7	C++中的文件 I/O	325
11.7.1	构造函数	325
11.7.2	文件打开操作	326
11.7.3	文件关闭操作	327
11.8	上机操作	327
11.8.1	C 文件操作	327
11.8.2	C++文件操作	330
11.9	小结	335
11.10	习题	335
习题答案		338
第 1 章答案		338
第 2 章答案		339
第 3 章答案		341
第 4 章答案		342
第 5 章答案		343
第 6 章答案		345
第 7 章答案		346
第 8 章答案		348
第 9 章答案		349
第 10 章答案		351
第 11 章答案		352

第1章 C语言及程序设计初步

本章学习目标

- ◆ 了解程序设计语言的发展历程与C语言产生的背景
- ◆ 掌握C语言的特点及C程序的基本构成
- ◆ 了解C程序的编辑、编译、连接和执行过程
- ◆ 熟悉 Visual C++ 6.0 集成开发环境

近年来,随着计算机技术的飞速发展,软件开发领域涌现出许多程序设计语言。据统计,自1954年第一种高级程序设计语言FORTRAN产生以来,已经出现了400多种语言。这些语言有些经过进化和发展成为更加流行的语言,而有些由于固有的缺陷而逐渐退出了历史的舞台,有些由于只应用于某个特定的领域而不为人们所熟知。

C语言作为第三代编程语言的代表,自产生以来,一直受到人们的重视和关注。C语言流行程度非常之广、影响非常之大。目前几乎所有大学的计算机课程中有关程序设计部分讲解的都是C语言,仅从这一点来看,C语言就会继续流行下去。

C语言兼具高级语言和低级语言的特点,非常适合编写系统程序,曾被誉为是真正的程序设计者的语言。尽管近些年来,随着面向对象技术的发展,C作为一种结构化程序设计语言已逐渐由面向对象的C++所替代,然而C语言是构成C++语言的基础,要想学好C++语言,就必须首先学习和掌握C语言。

另外,目前许多操作系统和编译系统都是由C语言开发而成,采用C语言进行程序设计可以更好地与这些系统相结合。例如Windows API函数的描述和调用就是采用与C语言相似的规则。因此,对于每个有志于进入程序设计领域的人来说,C语言将是其首选。

1.1 程序设计语言的发展

任何一种计算机系统都包括硬件(hardware)和软件(software)两大部分。硬件只是提供了计算的可能性,还必须有支持和管理计算机的软件,系统才能实现计算(computing)。从这个意义上讲,硬件是计算机的物质基础,而软件则是计算机的灵魂。

注意: 计算机领域的计算不仅仅指数值计算,在计算机中运行的所有指令都可以称为计算。例如,逻辑推理、事务处理、数据管理等。

软件的一个核心部分就是计算机编程语言。在计算机发展初期，所谓软件，就是程序。软件的设计和开发几乎是由单个程序员所完成的。

随着计算机技术的发展，传统的软件设计、开发和生产技术已不能适应信息时代的发展要求。于是计算机科学工作者将工程学的基本原理和方法引进软件设计和生产中，形成了一门新兴的学科，即软件工程学。

软件工程学出现后，软件的内含以及外延都发生了重大的变化，软件开发已不单纯地指程序设计，它囊括了系统分析和设计、文档编写、程序设计以及文档和程序版本管理等诸多方面。

尽管如此，程序设计和编码仍将是形成最终软件产品的必经步骤，而程序设计和编码则一刻也无法脱离计算机编程语言。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的发展历程。

1. 机器语言

计算机执行的指令实际上是 1 和 0 二进制数序列，它们代表计算机内部产生的电子信号。在计算机应用的最初十多年中，程序员主要使用计算机能够识别的二进制数来编写程序，这也是计算机唯一可以读懂的语言，一般称作机器语言。这种语言虽然十分简单，机器可以读懂，但对于程序员来说却很不方便。

完成一个简单的计算公式也要编写几十条指令，编程工作枯燥而繁琐，程序冗长而难读，调试、修改和维护都是难题。另外，采用机器语言编写的程序往往依赖于特定的硬件系统，可移植性很差。因此，早期的计算机应用不广，编程的专业性极强。机器语言通常被称为第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦，计算机专家开始尝试用人们比较习惯的符号来代替机器的二进制串指令，例如用 ADD 来代替 001 表示加法操作，用 MOV 来代替 010 表示数据移动等。

这样，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要使用一个专门的程序来负责将这些符号翻译成二进制数的机器语言，这种翻译程序称为汇编程序。

使用汇编语言编程比使用机器语言编程要容易得多。另外，由于汇编语言指令与机器语言指令基本上为一对一或一对多，可见汇编系统的程序开发也不太复杂，因此汇编语言编程很快取代了机器语言编程。

汇编语言和机器语言都属于低级语言，而且它们的语言结构都以面向机器的指令序列形式为主，与人的习惯语言方式距离较远。采用汇编语言编写的程序同样存在十分依赖于机器的问题，移植性不好，程序代码冗长，不易于编写大规模程序，可读性和可维护性都较差。

3. 高级语言

从最初与计算机交流的痛苦经历中,人们逐渐意识到,应该设计一种更接近于数学语言或者人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用。经过努力,最早由 IBM 的 John Backus 领导的研究小组于 1954 年提出了一个高级语言规范 IBM FORTRAN (Mathematical Formula Translation System)。3 年后 FORTRAN 1.0 版及其编译系统正式对外公布。

提示: 由汇编语言和高级语言编写的程序,均需要通过相应的翻译系统翻译成机器语言。有些语言的翻译过程是对源程序逐行解释,而有些语言一次翻译所有的源代码。逐行解释程序设计语言需要通过解释程序完成对源代码的翻译,而一次翻译所有源代码的程序设计语言则通过编译程序完成翻译。

FORTRAN 语言的诞生是计算机技术发展的一个里程碑。在此后的 40 多年时间里,共产生了近百种程序设计语言,形成了一种百花齐放的局面。其中, FORTRAN、ALGOL、COBAL、Pascal、BASIC、Ada、LISP 等都是影响较大的高级程序语言。

科学技术的发展永远不会停止不前,高级程序设计语言也是如此。从 FORTRAN 语言开始,高级程序设计语言主要在教育领域的扩展和软件设计方法的改进两个方面向前发展,这两个方向既相对独立又互相影响。

在教育领域的扩展方面,COBAL 语言的产生起非常重要的作用。COBAL 语言使计算机应用领域从单纯的科学计算向商业和事务处理迈出了关键性的一步,现在的数据库技术也正是起源于此。

在软件设计思想和设计方法发展这条主线上,ALGOL 语言的设计思想对后来的程序设计语言产生了很大的影响。目前许多高级程序设计语言都直接或间接起源于 ALGOL 语言。

随着软件的种类、规模和复杂性的增加,软件的可靠性和可维护性问题日益突出。20 世纪 60 年代在软件开发领域出现了一个名词——软件危机,该词非常形象地描述了那时软件开发、生产当中所面临的难题。为了解决软件危机问题,人们提出以系统的观点来对待软件设计和开发,从而产生了一些成熟的软件设计方法和程序设计语言。

1969 年,结构化的程序设计方法被提出;1970 年,第一个结构化的程序设计语言——Pascal 语言出现。20 世纪 80 年代初,面向对象的程序设计(OOP)思想产生,其中 C++ 语言就是面向对象程序设计语言的典型代表。

1.2 C 语言的产生

C 语言是由贝尔(Bell)实验室的 Dennis Ritchie 于 1972 年开发出来,它是一种结构化程序设计语言。C 语言源于 Ken Thomapson 的 B 语言,而 B 语言又源自于 BCPL (Basic Combined Programming Language) 语言,该语言由 Martin Richards 所开发。实际上, C 语

言的前身就是对程序设计思想产生重要影响的 ALGOL 语言, 它于 60 年代早期由国际委员会设计和开发。

此后, 由伦敦和剑桥大学合作开发的 CPL (Combined Programming Language) 语言对 ALGOL 进行了改进, 使其能够直接做较低层次的操作。然而, 跟 ALGOL 语言一样, CPL 语言有太多的特性, 很难理解而且实现起来也很困难, 为解决这一问题 BPCL 语言应运而生。表 1-1 反映了 C 语言的发展历史。

表 1-1 C 语言发展历史

语言	时间	开发者
ALGOL	20 世纪 60 年代早期	国际委员会
CPL	1963	伦敦和剑桥大学
BCPL	1967	剑桥大学的 Martin Richards
B	1970	贝尔实验室的 Ken Thomapson
C	1972	贝尔实验室的 Dennis Ritchie

最初, C 语言是在 UNIX 操作系统上开发的, 此后出现了许多 C 语言编译和集成工具, 这些工具在源程序水平上大都能够很好地兼容。然而, 由于没有统一的标准, 这些工具之间也有不一致的地方。

为了改变这种状况, 美国国家标准协会 (ANSI) 于 1983 年制定了 ANSI C 标准。至此, 所有的 C 语言工具都遵从这个标准, 只要编写符合 ANSI 标准的 C 语言源代码, 就可以保证在这些工具中很好地通用。

1.3 C 语言的特点

C 语言产生的背景以及 C 语言设计者的初衷, 决定了 C 语言具有许多明显不同于其他程序设计语言的特点。多年来, C 语言一直经久不衰并不断进化和发展, 完全得益于它的这些特点。

1. C 语言兼具低级语言和高级语言的特性

C 语言具有允许直接对位、字节和字进行操作的特点。C 语言所提供的指针类型使其具有寻址内存区域的能力, 而且 C 语言具有与汇编语言的透明接口。在 C 语言中, 可以方便地插入汇编语言从而使程序达到最高的效率。C 语言的这种低级语言的特点使其非常适合编写系统程序, Dennis Ritchie 正是基于此而开发 C 语言的。

C 语言像 Pascal、BASIC 等高级语言一样, 支持数据类型的概念。C 语言中定义了 5 种基本数据类型, 同时还允许程序员灵活地定义其他数据类型, 这极大地扩展了 C 语言的灵活性。