

普通高等教育“计算机类专业”规划教材

C程序设计与案例分析

刘志海 鲁青 主编

赵协广 王亮 高洁 王成龙 副主编



清华大学出版社

普通高等教育“计算机类专业”规划教材

C程序设计与案例分析

刘志海 鲁青 主编

赵协广 王亮 高洁 王成龙 副主编

清华大学出版社

北京

内 容 简 介

本书通过大量实例讲解 C 语言程序设计方法。全书共 12 章,首先介绍数据类型及表达式、三种基本结构的程序设计、数组、指针、结构体与链表和文件等内容;其次,特别安排了一章综合设计实例,通过万年历设计、通讯录设计、访问 dbf 数据库等 5 个综合实例培养读者分析问题和综合应用 C 语言基本知识解决问题的能力;最后,作为 C 语言与硬件联合应用的实例,介绍了 C 语言在开发 89C52 单片机中的应用,以提高读者的学习兴趣。各章均设有不同数量的应用实例和习题,内容讲解透彻。

本书附带电子教案、详细的习题参考答案和试题库管理系统,通过试题库系统可以快速输出规范正式的试卷和参考答案。

本书注重理论与实践的结合,融理论讲解、实例分析和实验指导为一体。本书可作为高等院校公共基础课教材或课程设计参考书,也适合于工程技术人员或 C 语言自学者使用。本书程序在 Visual C++ 6.0 环境下调试通过。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 程序设计与案例分析/刘志海,鲁青主编. —北京:清华大学出版社,2014

普通高等教育“计算机类专业”规划教材

ISBN 978-7-302-35959-3

I. ①C… II. ①刘… ②鲁… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 066038 号

责任编辑:白立军 战晓雷

封面设计:常雪影

责任校对:白 蕾

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:22.5

字 数:545 千字

版 次:2014 年 8 月第 1 版

印 次:2014 年 8 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

产品编号:053112-01

C 程序设计语言最早是由 Dennis Ritchie 于 1972 年设计并实现的,从那时起,C 语言即不断展现其青春活力和卓越功能,并风靡全球,成为世界上学习和应用最多的一门高级语言。许多软件,如 UNIX 操作系统、C 编译器和几乎所有的 UNIX 应用程序等,都是在 C 语言及其衍生的各种语言的基础上开发出来的。

本书从 C 语言的语法基础入手,由浅入深,用大量的实例讲解 C 语言程序的设计方法,每一章后都有一定数量的练习和编程习题,帮助读者掌握相关的知识点。本书主要内容包括 C 语言的数据类型、运算符和表达式、C 语言程序的基本结构、数组和指针、函数、结构体与链表、文件、综合程序设计、C 语言在单片机开发中的应用和实验指导。

本书具有以下特点:

(1) 每章均有若干个应用实例,类型多样,内容丰富,分析透彻,以便读者阅读理解和掌握。

(2) 特别安排了一章综合实例,通过 5 个综合实例,即万年历设计、大数字进制转换、彩票模拟程序、通讯录设计以及读取 dbf 数据表格,培养读者分析问题、设计算法和利用 C 语言编程解决实际问题的能力。

(3) 安排一章介绍 C 语言在单片机开发中的应用,以提高读者的学习兴趣,帮助读者开阔视野,了解 C 语言在硬件设计中的编程应用,精选当前流行的单片机开发练习板进行实例设计。

(4) 安排了一章实验指导,指导学生进行上机练习。

(5) 安排 2012 年 3 月和 9 月两套全国计算机等级考试二级 C 笔试试卷,供读者测试和练习。

(6) 本书附带电子课件、源程序、习题参考答案以及自主知识产权的试题库管理系统,教师可以直接使用试题库管理系统产生正式的考试试卷及参考答案。

(7) 本书是作者在十余年的教学和编程应用实践的基础上,并综合多位同行的教学科研经验精心编写而成的。

对于理论教学 30 学时的专业,建议学时分配如下:第 1 章 C 语言概述 2 学时;第 2 章数据类型 2 学时;第 3 章运算符和表达式 2 学时;第 4 章 C 语言程序的基本结构 4 学时;第 5 章数组与指针 6 学时;第 6 章函数与参数传递 4 学时;第 7 章编译预处理 1 学时;第 8 章结构体与链表 4 学时;第 9 章文件 1 学时;第 10 章综合设计实例 2 学时;第 11 章 C 语言在单片机开发中的应用 2 学时。其他专业的授课学时,可以参照进行。

不同专业可以根据培养计划和教学大纲的要求,选讲本书第 8 章 8.4 节,第 10 章 10.4 节

和 10.5 节,第 11 章 11.2 节的内容。

本书由山东科技大学的刘志海、鲁青任主编,赵协广、王亮、高洁、王成龙任副主编。刘志海编写了本书的第 1、第 5 和第 8 章,王亮、高洁编写了本书的第 3 和第 4 章,王宝仁、武洪恩编写了本书的第 6 和第 9 章,王成龙编写了本书的第 2 章,鲁青编写了本书的第 12 章和附录,赵协广编写了本书的第 10 章,梁慧斌、李学华参与了本书第 7 和第 11 章的编写和校稿,机电控制与智能装备研究所的李守志、王天超、刘继龙、苏兴明、朱岩朋等研究生也参与了相关材料的整理和校稿,最后由刘志海进行了全书统稿。部分从事 C 语言教学的同事对本书的编写提出了许多合理的建议,在此对参与本书立项及撰写的有关同事同行表示感谢。

本书程序全部在 Visual C++ 6.0 环境下调试通过。本书配套的多媒体课件、实例源代码和习题参考答案可在清华大学出版社网站(www.tup.com.cn)下载,或发邮件至 zhiliu@126.com 与作者联系。

由于编写时间仓促及作者能力有限,书中难免存在不当之处,恳请读者批评指正。

作 者

2014 年 5 月

F O R E W O R D

第 1 章 C 语言概述	/1
1.1 计算机语言的发展	/1
1.1.1 机器语言	/1
1.1.2 汇编语言	/2
1.1.3 高级语言	/2
1.1.4 结构化程序设计语言	/2
1.2 C 语言的发展	/3
1.3 C 语言的特点及简单程序组成	/4
1.3.1 C 语言的特点	/4
1.3.2 C 语言程序的组成	/4
1.4 C 程序的调试方法	/7
1.5 简单程序设计入门	/8
1.6 Visual C++ 6.0 集成环境调试	/9
1.6.1 启动 Visual C++ 6.0	/9
1.6.2 源程序的调试与运行	/11
习题	/11
第 2 章 数据类型	/12
2.1 标识符和字符集	/12
2.1.1 标识符	/12
2.1.2 字符集	/13
2.2 数据类型	/14
2.3 常量和变量	/14
2.3.1 常量和符号常量	/14
2.3.2 变量	/15
2.4 整型数据类型	/15
2.4.1 整型常量	/15
2.4.2 整型变量	/16
2.5 浮点型数据类型	/19
2.5.1 浮点常量	/19
2.5.2 浮点变量	/20
2.5.3 单精度浮点型数据的存储	/21

2.6 字符型数据类型	/22
2.6.1 字符型常量	/22
2.6.2 字符型变量	/23
2.6.3 字符串型常量	/24
2.7 不同数据类型之间的转换	/24
2.7.1 自动转换	/25
2.7.2 强制类型转换	/25
习题	/26
第3章 运算符和表达式	/28
3.1 算术运算符和算术表达式	/28
3.1.1 算术运算符	/28
3.1.2 算术表达式	/30
3.2 赋值运算符和赋值表达式	/31
3.2.1 赋值运算符	/31
3.2.2 赋值表达式	/31
3.2.3 赋值语句	/32
3.3 关系运算符和关系表达式	/33
3.3.1 关系运算符	/33
3.3.2 关系表达式	/33
3.4 逻辑运算符和逻辑表达式	/34
3.4.1 逻辑运算符	/34
3.4.2 逻辑表达式	/35
3.5 条件运算符和条件表达式	/36
3.5.1 条件运算符	/36
3.5.2 条件表达式	/36
3.6 逗号运算符和逗号表达式	/37
3.6.1 逗号运算符	/37
3.6.2 逗号表达式	/37
习题	/38
第4章 C 语言程序的基本结构	/41
4.1 结构化程序设计方法与算法	/41

4.1.1	结构化程序设计方法	/41
4.1.2	算法	/43
4.2	C 程序语句	/47
4.3	顺序结构程序设计	/50
4.3.1	字符的输入和输出	/50
4.3.2	字符串的输入与输出	/52
4.3.3	格式化输入与输出	/54
4.4	选择结构的基本形式	/58
4.4.1	简单分支结构	/58
4.4.2	双分支结构	/59
4.4.3	多分支结构	/61
4.4.4	switch...case 分支结构	/63
4.5	选择结构的嵌套	/65
4.6	循环结构的基本形式	/68
4.6.1	if...goto 构成的循环	/68
4.6.2	while 循环	/69
4.6.3	do...while 循环	/72
4.6.4	for 循环	/74
4.7	循环结构的嵌套	/76
4.8	循环控制语句	/78
4.8.1	break 语句	/78
4.8.2	continue 语句	/79
4.9	实例	/80
	习题	/84
第 5 章	数组与指针	/87
5.1	一维数组	/87
5.1.1	数组的定义	/87
5.1.2	数组元素的引用	/89
5.1.3	数组的初始化	/90
5.1.4	一维数组的应用实例	/92
5.2	二维数组及多维数组	/96
5.2.1	二维数组的定义	/96

5.2.2	二维数组的引用	/97
5.2.3	二维数组的初始化	/97
5.2.4	二维数组的应用实例	/98
5.3	字符数组	/101
5.3.1	字符数组的定义	/102
5.3.2	字符数组的初始化	/102
5.3.3	字符数组的引用	/103
5.3.4	字符串	/103
5.4	指针变量和指针运算符	/105
5.4.1	地址与指针	/105
5.4.2	指针变量定义及指针运算	/107
5.4.3	指针变量的引用	/109
5.4.4	指针的运算	/110
5.4.5	C 语言中指针变量赋值的几种 错误方法	/112
5.5	指向数组的指针	/112
5.5.1	指针与一维数组	/113
5.5.2	指针与二维数组	/115
5.5.3	指针与字符串	/117
5.5.4	指针数组	/119
5.6	实例	/121
	习题	/125
第 6 章	函数与参数传递	/129
6.1	概述	/129
6.2	函数的定义与调用	/132
6.2.1	函数定义的一般形式	/132
6.2.2	函数的声明	/135
6.2.3	函数的调用	/136
6.2.4	形式参数与实际参数	/137
6.2.5	函数的返回值	/140
6.2.6	函数调用时参数间的传递	/141
6.3	函数的嵌套调用与递归调用	/142

6.3.1	函数的嵌套调用	/142
6.3.2	函数的递归调用	/145
6.4	常用的数值和字符串处理函数	/150
6.4.1	数值处理函数	/150
6.4.2	字符串处理函数	/151
6.5	变量的作用域和存储类型	/156
6.5.1	局部变量	/156
6.5.2	全局变量	/158
6.5.3	变量的存储类别	/160
6.6	指针作为函数的参数	/165
6.7	指向函数的指针	/167
6.7.1	函数的指针	/167
6.7.2	用指向函数的指针作函数 参数	/169
6.8	返回指针的函数	/171
6.9	main 函数中的参数	/173
6.10	实例	/175
习题		/178
第 7 章	编译预处理	/183
7.1	宏定义	/183
7.1.1	无参宏定义	/183
7.1.2	带参宏定义	/187
7.2	文件包含	/189
7.3	条件编译	/190
7.3.1	#ifdef 命令	/190
7.3.2	#ifndef 命令	/192
7.3.3	#if 命令	/192
7.4	实例	/193
习题		/194
第 8 章	结构体与链表	/196
8.1	结构体的定义和引用	/196

8.1.1	结构体类型定义	/196
8.1.2	结构体类型变量的定义	/198
8.1.3	结构体变量的初始化和成员 引用	/200
8.2	结构体数组	/204
8.2.1	结构体数组的定义	/204
8.2.2	结构体数组的初始化	/205
8.3	指向结构体的指针	/208
8.3.1	结构体指针变量的定义	/208
8.3.2	结构体指针变量的赋值	/209
8.3.3	结构体指针变量成员的引用 ...	/209
8.3.4	指向结构体数组的指针	/210
8.3.5	结构体指针数组	/211
8.3.6	结构体变量和结构体指针作为 函数的参数	/211
8.4	链表的基本操作	/215
8.4.1	单链表	/215
8.4.2	内存操作函数	/216
8.4.3	单链表的基本操作	/217
8.5	共用体的定义和引用	/229
8.5.1	共用体类型及变量的定义	/229
8.5.2	共用体变量的引用方法	/230
8.5.3	共用体变量的赋值	/231
8.5.4	共用体类型数据的特点	/232
8.6	typedef 定义类型	/233
8.6.1	用于对数据类型的命名	/233
8.6.2	用于对数组和指针类型的 命名	/234
8.6.3	typedef 与 #define	/235
习题		/235
第9章	文件	/237
9.1	文件概述	/237

9.1.1	文件的基本概念	/237
9.1.2	文件的分类	/237
9.1.3	文件的基本操作	/238
9.2	文件类型指针	/238
9.3	文件的打开与关闭	/239
9.3.1	文件的打开	/240
9.3.2	文件的关闭	/241
9.4	文件读写	/242
9.4.1	读字符函数 fgetc	/242
9.4.2	写字符函数 fputc	/244
9.4.3	写字符串函数 fputs	/245
9.4.4	读字符串函数 fgets	/246
9.4.5	格式化读写函数 fscanf 和 fprintf	/247
9.4.6	数据块读写函数 fread 和 fwrite	/249
9.5	文件的定位	/251
9.5.1	随机定位函数 fseek	/251
9.5.2	文件头定位函数 rewind	/252
9.5.3	当前读写位置函数 ftell	/252
9.6	文件检测函数	/252
9.6.1	文件结束检测函数 feof	/252
9.6.2	读写文件出错检测函数 ferror	/252
9.6.3	文件出错标志和文件结束标志置 0 函数 clearerr	/253
9.7	文件应用举例	/254
习题		/257
第 10 章	综合设计实例	/258
10.1	万年历设计	/258
10.1.1	功能要求	/258
10.1.2	算法分析	/258
10.1.3	函数介绍	/259

10.1.4	参考程序	/259
10.1.5	运行结果	/261
10.2	大数字进制转换	/261
10.2.1	功能要求	/261
10.2.2	函数设计	/261
10.2.3	程序示例	/262
10.2.4	程序验证	/265
10.3	彩票模拟程序	/265
10.3.1	功能要求	/265
10.3.2	总体设计	/266
10.3.3	函数设计	/266
10.3.4	程序代码	/267
10.3.5	测试结果	/271
10.4	简单通讯录设计	/271
10.4.1	功能要求	/271
10.4.2	总体设计	/272
10.4.3	存储结构	/272
10.4.4	函数设计	/272
10.4.5	程序示例	/274
10.4.6	测试结果	/279
10.5	读取 dbf 数据表格	/280
10.5.1	dbf 表文件的结构	/281
10.5.2	读取 dbf 表的内容	/282
	习题	/284
第 11 章 C 语言在单片机开发中的应用		/285
11.1	位运算	/285
11.1.1	“按位与”运算	/286
11.1.2	“按位或”运算	/287
11.1.3	“按位异或”运算	/288
11.1.4	“求反”运算	/289
11.1.5	“左移”运算	/289
11.1.6	“右移”运算	/289

11.1.7 位复合赋值运算	/290
11.2 89C52 单片机 C 语言应用	/290
11.2.1 STC89C52RC 单片机介绍	/290
11.2.2 Keil μ Vision2 单片机开发 环境	/291
11.2.3 STC-ISP 软件介绍	/295
11.2.4 C51 单片机开发板介绍	/297
11.2.5 C 语言单片机应用	/299
第 12 章 实验指导	/306
实验一 Visual C++ 6.0 集成环境调试	/306
实验二 数据类型、运算符和表达式	/307
实验三 顺序程序设计	/309
实验四 选择结构程序设计	/310
实验五 循环控制	/311
实验六 数组	/312
实验七 指针	/312
实验八 函数	/314
实验九 预处理	/315
实验十 结构体与共用体	/315
实验十一 文件、位操作	/316
附录 A 运算符的优先级	/318
附录 B 常用字符与 ASCII 代码对照表	/320
附录 C 2012 年 3 月全国计算机等级考试二级 C 笔试试卷	/322
附录 D 2012 年 9 月全国计算机等级考试二级 C 笔试试卷	/332
参考文献	/343

第 1 章 C 语言概述

【本章概述】

C 语言是一种通用的程序设计语言,尤其适合于编写编译器和操作系统。C 语言除提供了很多数据类型之外,还提供了基本的控制流结构,如选择结构、循环结构等。在利用 C 语言进行程序设计之前,有必要了解 C 语言程序文件的组成以及 C 语言程序的编译和调试过程。

【学习要求】

- 了解: C 语言的发展。
- 掌握: C 语言的特点、简单 C 语言程序的组成。
- 掌握: C 语言程序的上机调试步骤。
- 重点: 简单 C 语言程序的组成和上机调试步骤。
- 难点: 集成开发环境和程序调试方法。

1.1 计算机语言的发展

自 1946 年第一台电子计算机问世以来,计算机已被广泛地应用于生产、生活的各个领域,推动着社会的进步与发展。特别是 Internet 出现后,传统的信息收集、传输及交换方式发生了革命性的改变。计算机科学的发展依赖于计算机硬件和软件技术的发展,硬件是计算机的躯体,软件是计算机的灵魂。没有软件,计算机只是一台“裸机”,什么也不能干;有了软件,计算机才有“思想”,才能做相应的事。软件是用计算机语言编写的。计算机语言的发展经历了从机器语言、汇编语言到高级语言的历程。

1.1.1 机器语言

计算机使用的是由 0 和 1 组成的二进制数,二进制编码方式是计算机语言的基础。计算机发明之初,科学家只能用二进制数编制的指令控制计算机运行。每一条计算机指令均由一组 0、1 数字按一定的规则排列组成,若要计算机执行一项简单的任务,需要编写大量的这种指令。这种由有规则的二进制数组成的指令集就是机器语言(machine language,也称为指令系统)。不同系列的 CPU 具有不同的机器语言,如目前个人计算机中常用 AMD 公司的系列 CPU 和 Intel 公司的系列 CPU 就具有不同的机器语言。

机器语言是计算机唯一能识别并直接执行的语言,与汇编语言或高级语言相比,执行效率高,但可读性差,不易记忆;编写程序既难又繁,容易出错;程序调试和修改难度巨大,不容易掌握和使用。此外,因为机器语言直接依赖于中央处理器,所以用某种机器语言编写的程序只能在相应的计算机上执行,无法在其他型号的计算机上执行,也就是说,可移植性差。

1.1.2 汇编语言

为了减轻使用机器语言编程的痛苦,20 世纪 50 年代初,出现了汇编语言(assembly language)。汇编语言用比较容易识别和记忆的助记符替代特定的二进制串。下面是几条 Intel 80x86 的汇编指令:

```
ADD AX,BX      ;表示将寄存器 AX 和 BX 中的内容相加,结果保存在寄存器 AX 中
SUB AX,NUM     ;表示将寄存器 AX 中的内容减去 NUM,结果保存在寄存器 AX 中
MOV AX,NUM     ;表示把数 NUM 保存在寄存器 AX 中
```

通过这种助记符,人们就能较容易地读懂程序,调试和维护也更方便了。但这些助记符号计算机无法识别,需要一个专门的程序将其翻译成机器语言,这种翻译程序被称为汇编程序。

汇编语言的一条汇编指令对应一条机器指令,汇编语言与机器语言在性质上是一样的,只是表示方式做了改进,与机器语言相比其可移植性仍然较差。总之,汇编语言是符号化的机器语言,执行效率稍逊于机器语言,由于采用了助记符,一定程度上提高了程序的可读性,因此,汇编语言至今仍是一种常用的软件开发工具。

1.1.3 高级语言

尽管汇编语言比机器语言方便,但汇编语言仍然具有许多不便之处,程序编写的效率远远不能满足需要。1954 年,第一个高级语言——FORTRAN 问世了。高级语言是一种利用能表达各种意义的“词”和“数学公式”按一定的“语法规则”编写程序的语言,也称为高级程序设计语言或算法语言。半个多世纪以来,有几百种高级语言问世,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、NOBOL、Pascal、C、PROLOG、C++、Visual C++、Visual Basic、Delphi 和 Java 等。高级语言的发展经历了从早期语言到结构化程序设计语言再到面向对象程序设计语言的过程。

高级语言与自然语言和数学表达式相当接近,不依赖于计算机型号,通用性较好。高级语言的使用,大大提高了程序编写的效率和程序的可读性。

与汇编语言一样,计算机无法直接识别和执行高级语言,必须翻译成等价的机器语言程序(称为目标程序)才能执行。高级语言源程序翻译成机器语言程序的方法有“解释”和“编译”两种。解释方法是边解释边执行,如早期的 BASIC 语言即采用解释方法,在执行 BASIC 源程序时,解释一条 BASIC 语句,执行一条语句。编译方法采用相应语言的编译程序,先把源程序编译成指定机型的机器语言目标程序,然后再把目标程序和标准库函数连接装配成完整的目标程序,在相应的机型上执行。如 C、C++、Visual C++ 及 Visual Basic 等语言均采用编译的方法。编译方法比解释方法效率更高。

1.1.4 结构化程序设计语言

高级语言编写程序的效率虽然比汇编语言高,但随着计算机硬件技术的日益发展,人们对大型、复杂的软件需求量剧增,而同时因缺乏科学规范、系统规划与测试,程序含有过多错误而无法使用,甚至带来巨大损失。20 世纪 60 年代中后期“软件危机”的爆发,使人们认识到大型程序的编制不同于小程序。“软件危机”的解决需要对程序设计方法、程序的正确性

和软件的可靠性等问题进行深入研究。

结构化程序设计(structural programming)是进行以模块功能和处理过程设计为主的详细设计的基本原则。其概念最早由 E. W. Dijkstra 在 1965 年提出的,是软件发展的一个重要的里程碑。结构化程序设计的主要观点是采用自顶向下、逐步求精及模块化的程序设计方法;使用顺序、选择和循环 3 种基本控制结构来构造程序,而模块化是把程序要解决的总目标分解为子目标,再进一步分解为具体的小目标,每一个小目标即称为一个模块。结构化程序设计主要强调的是程序的易读性。

结构化程序设计有以下优点:

- (1) 整体思路清楚,目标明确。经过自顶向下、逐步求精的分析后,目标变得十分明确。
- (2) 设计工作中阶段性非常强,有利于系统开发的总体管理和控制。软件设计分为可行性分析、系统分析、功能设计、代码设计、运行及维护多个阶段,每个阶段的任务非常明确,需要产生对应的技术或说明文档。
- (3) 在系统分析时可以诊断出原系统中存在的问题和结构上的缺陷。软件开发只是整个系统设计的一部分,经过可行性分析、功能设计和详细设计之后的开发目标已非常明确,系统开发中的问题也已基本考虑周全。

结构化程序设计也存在以下的缺点:

- (1) 用户要求难以在系统分析阶段准确定义,致使系统在交付使用时产生许多问题。
- (2) 用系统开发每个阶段的成果来进行控制,不能适应事物变化的要求。
- (3) 系统的开发周期长。软件设计的每个阶段都需要投入一定的时间去分析和设计,导致整个系统的开发周期长。

1.2 C 语言的发展

C 语言是世界上最流行的计算机程序设计语言之一。

C 语言的祖先是 BCPL 语言。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,设计出很简单且很接近硬件的 B 语言(取 BCPL 的首字母),他用 B 语言写了第一个 UNIX 操作系统。1972 年美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言,他取了 BCPL 的第二个字母作为这种语言的名字,这就是 C 语言。1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的 *The C Programming Language* 一书,奠定了 C 语言的基础,形成了 K&R 的 C 语言标准。但是,在该书中并没有定义完整的 C 语言标准。随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,1983 年,由美国国家标准学会(ANSI)在此基础上制定了 C 语言标准,通常称之为 ANSI C。1987 年,ANSI 又公布了新标准——87 ANSI C。1990 年,国际标准化组织(ISO)接受 87 ANSI C 为 ISO C 的标准(ISO 9899—1990),目前许多流行的 C 编译系统都是以该标准为基础的。1999 年,ISO 又对 C 语言标准进行了修订,在基本保留原有 C 语言特征的基础上,针对应用需要,添加了一些功能,尤其是 C++ 中的部分功能,命名为 C99 标准。