

普通高等学校计算机类规划教材

# C语言程序设计

主编 刘渝妍 丁海燕 何红玲



科学出版社



# C语言程序设计

第 1 章 绪论

1

内 容 简 介

普通高等学校计算机类规划教材

# C 语言程序设计

主 编 刘渝妍 丁海燕 何红玲

副主编 邱 莎 俞锐刚 陈韬伟 何 英

参 编 朱晓丽 孙建洪

科学出版社

元 00.00 · 价 宝

(地址:北京) (电话:010-64015000)

## 内 容 简 介

本书注重以“案例—C 程序语法—综合应用”的方式组织教材的内容，在编写结构上按内容的相关性进行组织，由浅入深、循序渐进，同时注意突出重点、分散难点。每章都附有小结、习题，便于读者学习。在案例和习题方面注重收集生活中的问题、有趣味性的问题。案例的编写围绕问题描述、问题分析、程序描述和程序运行结果等方面进行。

全书分为基础篇和提高篇：基础篇包括绪论、算法、用 C 语言编写程序、选择结构、循环结构、数据组织、函数初步和指针初步；提高篇包括函数与程序结构、动态组织数据、文件、指针进阶和位运算。既可满足初学者学习 C 语言的基本要求，也可满足读者进一步进行程序设计的要求。

本书可作为普通高等院校计算机及相关专业的学生学习 C 语言程序设计的教材或教学参考书，也适合参加计算机等级考试的考生学习，同时还可以作为工程技术人员学习 C 语言的自学用书。

### 图书在版编目(CIP)数据

C 语言程序设计 / 刘渝妍, 丁海燕, 何红玲主编. —北京: 科学出版社, 2015.2  
普通高等学校计算机类规划教材

ISBN 978-7-03-043204-9

I. ①C… II. ①刘… ②丁… ③何… 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(2015)第 021154 号

责任编辑: 于海云 / 责任校对: 桂伟利

责任印制: 霍 兵 / 封面设计: 迷底书装

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>

文林印务有限公司印刷

科学出版社发行 各地新华书店经销

\*

2015 年 2 月第 一 版 开本: 787×1092 1/16

2015 年 2 月第一次印刷 印张: 19

字数: 450 000

定价: 40.00 元

(如有印装质量问题, 我社负责调换)

# 普通高等学校计算机类规划教材

## 编 委 会

### 总 主 编

李 彤 柳 青

### 主任委员

李 彤 云南大学软件学院

### 副主任委员

杨志军 云南省教育科学研究院

王 锋 昆明理工大学信息工程与自动化学院

柳 青 云南大学软件学院

### 委 员 (以姓名笔画为序)

王 新 云南民族大学数学与计算机科学学院

申时凯 昆明学院信息技术学院

杨林楠 云南农业大学基础与信息工程学院

杨朝凤 保山学院信息学院

余正涛 昆明理工大学信息工程与自动化学院

余建坤 云南财经大学信息学院

狄光智 西南林业大学计算机与信息学院

张学杰 云南大学信息学院

段利华 大理学院

徐天伟 云南师范大学信息学院

徐庆生 楚雄师范学院计算机科学系

解季萍 云南广播电视大学招生与系统建设处

# 前 言

C 语言是目前使用较多的程序设计语言之一。编者在多年的教学实践后,编写了《C 语言程序设计》教材。本书具有以下特点:

(1) 本书按 C 语言内容的难度分为基础篇和提高篇。基础篇包括:绪论、算法、用 C 语言编写程序、选择结构、循环结构、数据组织、函数初步和指针初步,构成了完整的 C 语言体系,可满足初学者学习 C 语言的基本要求。提高篇包括:函数与程序结构、动态组织数据、文件、指针进阶和位运算,从深度上满足读者进一步学习 C 语言,并进行程序设计和小型软件开发的需求。

(2) 本书的编写体系改变了目前以章节、知识点和语言体系组织内容的框架。如,第 6 章数据组织和第 10 章动态组织数据、第 7 章函数初步和第 9 章函数与程序结构、第 8 章指针初步和第 12 章指针进阶等,符合读者的认知过程,内容不断深入,一步一步引导读者进行学习,反映程序设计特有的思维方法。

(3) 本书以案例为驱动,选用了大量的案例。书中每节的内容均以“案例—C 程序语法—综合应用”为主线。每个案例的编写以“问题描述—问题分析—程序描述—程序运行结果—程序分析”进行组织。书中案例既完整,规模又不太大,并加以注释,便于读者理解和学习。

(4) 本书注重引入计算思维。程序设计不仅仅是语言、语法的罗列,更重要的是强调解决问题的思维过程,给出问题的解决方案。本书摒弃目前程序设计中以语法为重点,强调对问题的分析,使读者不再局限于单纯地了解和掌握 C 语言的基本语法规范,而是引导读者学习编写程序的方法。

(5) 本书同时有配套的《C 语言程序设计习题解答与上机指导》。根据书的章节内容,对每章的习题给出解析及参考答案。对应的上机指导,选择了目前最合适的开发工具(环境) Visual C++ 进行介绍,每个实验的编写以“实验基础—实验目的一实验内容—实验总结”进行组织。

全书由刘渝妍(第 2 章)、丁海燕(第 3 章、第 4 章)、何红玲(第 10 章、第 12 章)、邱莎(第 1 章)、俞锐刚(第 8 章)、陈韬伟(第 11 章、第 13 章)、何英(第 7 章、第 9 章)、朱晓丽(第 6 章、附录 A~E)、孙建洪(第 5 章)共同编写,并由刘渝妍、丁海燕进行统稿。

由于作者水平有限,书中若有不妥之处,敬请读者批评指正。

编 者

2014 年 10 月

# 目 录

## 基 础 篇

前言	1	第 1 章 绪论	2
1.1 引言	2	1.1 引言	2
1.2 C 语言出现的历史背景	4	1.2 C 语言出现的历史背景	4
1.2.1 C 语言的诞生	4	1.2.1 C 语言的诞生	4
1.2.2 C 语言标准的演变	5	1.2.2 C 语言标准的演变	5
1.3 C 语言的特点	6	1.3 C 语言的特点	6
1.3.1 基本特性	6	1.3.1 基本特性	6
1.3.2 优点	6	1.3.2 优点	6
1.3.3 缺点	7	1.3.3 缺点	7
1.4 一个简单的 C 语言程序	8	1.4 一个简单的 C 语言程序	8
1.5 程序与程序设计语言	11	1.5 程序与程序设计语言	11
1.5.1 程序与指令	11	1.5.1 程序与指令	11
1.5.2 程序设计语言	13	1.5.2 程序设计语言	13
1.5.3 程序的编辑与处理	14	1.5.3 程序的编辑与处理	14
1.6 运行 C 语言程序的步骤和方法	15	1.6 运行 C 语言程序的步骤和方法	15
1.6.1 运行 C 语言程序的步骤	15	1.6.1 运行 C 语言程序的步骤	15
1.6.2 运行 C 语言程序的方法	15	1.6.2 运行 C 语言程序的方法	15
1.7 程序设计中的命名规则	16	1.7 程序设计中的命名规则	16
本章小结	18	本章小结	18
习题 1	19	习题 1	19
第 2 章 算法	21	第 2 章 算法	21
2.1 算法的概念	21	2.1 算法的概念	21
2.1.1 算法的定义及特征	22	2.1.1 算法的定义及特征	22
2.1.2 算法的评价	23	2.1.2 算法的评价	23
2.2 算法的描述	24	2.2 算法的描述	24
2.2.1 自然语言描述算法	24	2.2.1 自然语言描述算法	24
2.2.2 流程图描述算法	26	2.2.2 流程图描述算法	26
2.2.3 伪代码描述算法	27	2.2.3 伪代码描述算法	27
2.3 简单算法举例	28	2.3 简单算法举例	28
2.4 计算思维	30	2.4 计算思维	30
2.4.1 计算思维的定义	30	2.4.1 计算思维的定义	30
2.4.2 计算思维的特征	32	2.4.2 计算思维的特征	32
2.5 结构化程序设计方法	32	2.5 结构化程序设计方法	32
2.5.1 自顶向下、逐步求精	33	2.5.1 自顶向下、逐步求精	33
2.5.2 基本控制结构	33	2.5.2 基本控制结构	33
2.5.3 模块化设计	33	2.5.3 模块化设计	33
本章小结	35	本章小结	35
习题 2	35	习题 2	35
第 3 章 用 C 语言编写程序	37	第 3 章 用 C 语言编写程序	37
3.1 C 语言程序的简单结构	37	3.1 C 语言程序的简单结构	37
3.1.1 程序的基本结构和格式	39	3.1.1 程序的基本结构和格式	39
3.1.2 C 语言的标识符	39	3.1.2 C 语言的标识符	39
3.2 常量、变量与赋值	40	3.2 常量、变量与赋值	40
3.2.1 常量	40	3.2.1 常量	40
3.2.2 变量与内存	41	3.2.2 变量与内存	41
3.2.3 变量的定义与赋值	43	3.2.3 变量的定义与赋值	43
3.3 运算符与表达式	44	3.3 运算符与表达式	44
3.3.1 算术运算符	44	3.3.1 算术运算符	44
3.3.2 赋值运算符	45	3.3.2 赋值运算符	45
3.3.3 自增、自减运算符	46	3.3.3 自增、自减运算符	46
3.3.4 条件运算符	46	3.3.4 条件运算符	46
3.3.5 逗号运算符	47	3.3.5 逗号运算符	47
3.3.6 求字节运算符	47	3.3.6 求字节运算符	47
3.4 基本输入输出	47	3.4 基本输入输出	47
3.4.1 格式输入输出	47	3.4.1 格式输入输出	47
3.4.2 字符输入输出	56	3.4.2 字符输入输出	56
3.5 基本数据类型	57	3.5 基本数据类型	57
3.5.1 整型数据在内存中的存储	58	3.5.1 整型数据在内存中的存储	58
3.5.2 浮点型数据在内存中的存储	59	3.5.2 浮点型数据在内存中的存储	59
3.5.3 字符型数据在内存中的存储	59	3.5.3 字符型数据在内存中的存储	59
3.6 各类数值型数据间的混合运算	60	3.6 各类数值型数据间的混合运算	60
3.6.1 运算中不同类型数据间的类型转换	60	3.6.1 运算中不同类型数据间的类型转换	60
3.6.2 数据类型的强制转换	61	3.6.2 数据类型的强制转换	61

本章小结 .....	61	6.2.3 二维数组的初始化 .....	106
习题 3 .....	62	6.2.4 二维数组编程 .....	107
<b>第 4 章 选择结构</b> .....	<b>66</b>	<b>6.3 字符数组</b> .....	<b>109</b>
4.1 关系运算符和关系表达式 .....	66	6.3.1 字符数组的定义 .....	109
4.2 逻辑运算符和逻辑表达式 .....	66	6.3.2 字符数组的初始化 .....	110
4.3 if 语句 .....	67	6.3.3 字符数组的输入和输出 .....	111
4.3.1 简单猜数游戏 .....	67	6.3.4 其他字符串处理函数 .....	114
4.3.2 if 单分支语句 .....	68	6.3.5 字符数组编程 .....	114
4.3.3 if 双分支语句 .....	69	<b>6.4 结构体变量</b> .....	<b>115</b>
4.3.4 if 多分支语句 .....	69	6.4.1 结构体类型的声明 .....	116
4.4 switch 语句 .....	70	6.4.2 定义结构体变量 .....	117
4.4.1 简单菜单程序 .....	70	6.4.3 结构体类型变量的初始化和引用 .....	118
4.4.2 switch 语句 .....	72	6.4.4 结构体数组 .....	119
本章小结 .....	73	6.4.5 结构体数组编程 .....	120
习题 4 .....	74	<b>6.5 共用体</b> .....	<b>121</b>
<b>第 5 章 循环结构</b> .....	<b>79</b>	6.5.1 共用体的概念 .....	123
5.1 求 $1+2+3+\dots+100$ .....	79	6.5.2 共用体类型的定义 .....	123
5.1.1 while 循环语句 .....	79	6.5.3 共用体变量的定义 .....	124
5.1.2 do...while 循环语句 .....	80	6.5.4 共用体变量的引用 .....	124
5.1.3 for 循环语句 .....	82	6.5.5 共用体类型数据的特点 .....	125
5.1.4 goto 语句以及用 goto 构成的循环 .....	83	<b>6.6 枚举类型</b> .....	<b>126</b>
5.2 判断素数 .....	83	本章小结 .....	127
5.2.1 break 语句 .....	83	习题 6 .....	127
5.2.2 continue 语句 .....	85	<b>第 7 章 函数初步</b> .....	<b>131</b>
5.3 嵌套循环 .....	87	7.1 计算组合数 $C_m^n$ .....	131
5.4 循环结构程序设计 .....	89	7.1.1 问题提出 .....	131
本章小结 .....	91	7.1.2 函数的分类 .....	132
习题 5 .....	92	7.1.3 函数的定义 .....	132
<b>第 6 章 数据组织</b> .....	<b>98</b>	7.1.4 函数的调用 .....	133
6.1 一维数组的定义和引用 .....	98	7.2 显示一条横线 .....	135
6.1.1 一维数组的定义 .....	98	7.2.1 不返回结果的函数应用 .....	135
6.1.2 一维数组元素的引用 .....	99	7.2.2 不返回结果的函数定义 .....	136
6.1.3 一维数组的初始化 .....	99	7.3 变量的作用域和生存期 .....	136
6.1.4 一维数组编程 .....	101	7.3.1 局部变量和全局变量 .....	136
6.2 二维数组的定义和引用 .....	104	7.3.2 变量的生存周期和存储属性 .....	138
6.2.1 二维数组的定义 .....	105	7.4 函数应用程序设计 .....	141
6.2.2 二维数组的引用 .....	106	本章小结 .....	143
		习题 7 .....	143



第 8 章 指针初步	149
8.1 指针基本概念	149
8.1.1 交换两个变量的值	149
8.1.2 地址和指针的概念	151
8.1.3 指针变量定义	151
8.1.4 指针变量的引用	152
8.1.5 指针作为函数的参数	153
8.2 指针与数组	153
8.2.1 冒泡排序法	153
8.2.2 指针、数组和地址的关系	155
8.2.3 数组名作为函数参数	157
8.3 指针与字符串	157
8.3.1 字符串的分类统计	157
8.3.2 字符串和字符指针	159
8.3.3 常用字符串处理函数	160
8.4 指针与结构体	163
8.4.1 制造虚拟汽车	163
8.4.2 指针与结构体	164
8.4.3 指针与结构体数组	165
8.5 指针与函数	165
8.5.1 简单模式匹配	165
8.5.2 定义函数返回值为指针	167
8.6 指针应用程序设计	169
本章小结	169
习题 8	169

## 提 高 篇

第 9 章 函数与程序结构	173
9.1 嵌套调用求组合数	173
9.2 函数递归调用	176
9.2.1 函数递归调用基本概念	176
9.2.2 汉诺塔问题	178
9.3 编译预处理	180
9.3.1 宏定义	181
9.3.2 文件包含	182
9.3.3 条件编译	182
9.4 学生成绩管理系统设计	185
9.4.1 分模块设计学生成绩管理系统	185
9.4.2 程序文件模块图	185
9.4.3 文件模块间的通信	185
本章小结	188
习题 9	189
第 10 章 动态组织数据	194
10.1 用链表构造学生信息	194
10.1.1 链表的概念	196
10.1.2 处理动态链表所需的函数	197
10.1.3 建立动态链表	198
10.2 输出学生链表信息	200
10.2.1 链表的遍历	200
10.2.2 链表遍历的简单应用	202
10.3 在链表中插入一个新学生信息	202
10.4 在链表中删除一个指定学生信息	206
10.5 动态组织数据综合应用	210
10.5.1 用 typedef 命名已有类型	210
10.5.2 综合应用	211
本章小结	214
习题 10	215
第 11 章 文件	222
11.1 C 文件的概念	222
11.1.1 文本文件和二进制文件	223
11.1.2 文件缓冲区	223
11.1.3 文件类型指针	224
11.2 文件的打开与关闭	225
11.2.1 文件的打开(fopen 函数)	225
11.2.2 文件的关闭(fclose 函数)	227
11.3 C 文件的读写	227
11.3.1 字符读写函数 fgetc 和 fputc	228
11.3.2 字符读写函数 fgets 和 fputs	232
11.3.3 数据块读写函数 fread 和 fwrite	235
11.3.4 格式化读写函数 fprintf 和 fscanf	237

11.4	C 文件的随机读写	239	12.3.2	定义指向指针数据的指针变量	258
11.4.1	文件定位	240	12.3.3	指向指针数据的指针变量	
11.4.2	随机读写	240		作为函数参数	259
11.5	C 文件的检测函数	241	12.3.4	命令行参数	260
11.5.1	文件结束检测函数	241	12.4	指针综合应用	261
11.5.2	读写文件出错检测函数	242		本章小结	263
11.5.3	文件出错标志和文件结束标志置 0 函数	242		习题 12	264
	本章小结	243	<b>第 13 章 位运算</b>		269
	习题 11	243	13.1	位运算符	269
<b>第 12 章 指针进阶</b>		247	13.1.1	按位与运算	269
12.1	用矩形法求多个函数的定积分	247	13.1.2	按位或运算	271
12.1.1	指向函数指针的概念	249	13.1.3	按位异或运算	271
12.1.2	指向函数的指针变量	249	13.1.4	求反运算	273
12.1.3	指向函数指针的调用	249	13.1.5	左移运算	273
12.2	查找关键字符号表	251	13.1.6	右移运算	274
12.2.1	指针数组的定义	252	13.1.7	不同长度的数据进行位运算	275
12.2.2	指针数组与二维数组	254	13.2	位运算应用与举例	275
12.3	用指向指针数据的指针实现字符串的排序	257	13.3	位域	277
12.3.1	指向指针数据的指针变量	258		本章小结	279
	习题 13	279		习题 13	279
<b>附录</b>		283	<b>附录 C 运算符的优先级和结合性</b>		285
附录 A	常用字符与 ASCII 代码对照表	283	<b>附录 D C 常用库函数</b>		286
附录 B	关键字及其用途	284	<b>附录 E VC6.0 常见编译错误信息</b>		290
<b>参考文献</b>		294			

本书主要介绍... 绪论... 本书主要介绍...

# 基础篇



# 第 1 章 绪 论

## 本章导读

C 语言是目前使用较多的程序设计语言之一。本章将由对 C 语言的基本应用情况、发展历史的介绍开始,并简要描述其主要特点,进而通过一个简单的程序案例说明 C 程序的主要结构、组成及源程序的处理过程,同时也对 C 程序设计中涉及的命名规则进行简介。最后通过一个具体编程实例介绍以 Microsoft Visual C++ 6.0 为运行平台的 C 程序编辑和运行过程。目的是使读者通过对本章的阅读,能对 C 语言有初步的了解。

## 1.1 引言

随机进行一个民意调查:您知道或听说过哪些计算机编程语言?得到的答案会是五花八门,C、Pascal、Basic、Fortran、C++、Java……这一系列编程语言的名称,就如同繁星点点,在程序设计的浩瀚天空中散发着璀璨的光芒。而这其中,C 语言无疑是被提及最多的最亮的一颗,它就如同北极星,引导着程序设计前进发展的方向。

C 语言是使用最广泛的语言之一,可以说,C 语言的诞生是现代程序语言革命的起点,是程序设计语言发展史中的一个里程碑。自 C 语言出现后,以 C 语言为根基的 C++、Java 和 C#等面向对象语言相继诞生,并在各自领域大获成功。但今天 C 语言依旧在系统编程、嵌入式编程等领域占据着统治地位。

根据 TIOBE<sup>①</sup>编程语言排行榜 2014 年 8 月公布的数据,如图 1.1~图 1.3 所示,C 语言高居榜首,多年来一直位于选用率榜单前列。

这样一种诞生于 20 世纪 70 年代的“古老”编程语言,至今仍为众人所追捧,几乎成为所有编程初学者的入门语言,同时也是很多编程高手的神功利器,焕发着蓬勃的生机。这是为何?

2012 年 6 月 30 日,O'Reilly<sup>②</sup>采访了 *Head First C*<sup>③</sup>一书的联合作者 David Griffiths,谈论了为什么今天还要学 C 语言,谈到以下几点:

- 
- ① TIOBE 是一个流行编程语言排行,每月更新,对编程语言的选择有一定的指导意义。排名权重基于世界范围内工程师数量、课程数量和第三方供应商数量。Google、Bing、Yahoo!、Wikipedia、Amazon、Youtube 和百度这些主流的搜索引擎,也将作为排名权重的参考指标。(声明:TIOBE 排名仅是数据统计的一个参考,既无关最好的编程语言,也无关被书写了最多行代码的编程语言。网址:www.tiobe.com。)
  - ② O'Reilly Media,Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司,同时是联机出版的先锋。从最畅销的 *The Whole Internet User's Guide & Catalog*(被纽约公共图书馆评为 20 世纪最重要的 50 本书之一)到 GNN(最早的 Internet 门户和商业网站),再到 WebSite(第一个桌面 PC 的 Web 服务器软件),O'Reilly Media,Inc.一直处于 Internet 发展的最前沿。(注:该文字引自百度百科。)
  - ③ 著名的 C 语言著作,图文并茂。来自 David Griffiths 和 Dawn Griffiths 夫妇的 *Head First C* 将会带你进入一个与众不同的 C 语言世界:这里没有令人乏味的“学究腔”,没有冗长枯燥的老生常谈,也没有令人望而却步的大堆练习。作者用诙谐、幽默的口吻为你讲述了一个又一个关于 C 的故事。

Aug 2014	Aug 2013	Change	Programming Language	Ratings	Change
1	2	▲	C	16,401%	+0.43%
2	1	▼	Java	14,984%	-0.99%
3	4	▼	Objective-C	9,552%	+1.47%
4	3	▼	C++	4,695%	-4.68%
5	7	▲	Basic	3,635%	-0.24%
6	6		C#	3,409%	-2.71%
7	8	▲	Python	3,121%	-0.48%
8	5	▼	PHP	2,864%	-3.83%
9	11	▲	Perl	2,218%	+0.18%
10	9	▼	JavaScript	2,172%	+0.08%
11	-	▲	Visual Basic	2,014%	+2.01%
12	13	▲	Visual Basic .NET	1,310%	-0.01%
13	10	▼	Ruby	1,242%	-0.83%
14	23	▲	F#	1,096%	+0.56%
15	18	▲	Pascal	1,044%	+0.42%
16	12	▼	Transact-SQL	1,043%	-0.35%
17	38	▲	ActionScript	1,001%	+0.69%
18	14	▼	Delphi/Object Pascal	0,915%	-0.00%
19	16	▼	Lisp	0,828%	+0.08%
20	17	▼	PL/SQL	0,786%	+0.03%

图 1.1 TIOBE 编程语言排行榜(2014 年 8 月)

Programming Language 2014 2009 2004 1999 1994 1989

C	1	2	2	1	1	1
Java	2	1	1	16	-	-
Objective-C	3	33	41	-	-	-
C++	4	3	3	2	2	2
C#	5	7	7	21	-	-
PHP	6	4	6	-	-	-
Python	7	6	8	26	22	-
JavaScript	8	9	9	18	-	-
Visual Basic .NET	9	-	-	-	-	-
Transact-SQL	10	29	-	-	-	-
Lisp	13	20	14	11	6	3
(Visual) Basic	87	5	5	3	3	7

图 1.2 TIOBE 主流编程语言历史排行榜

TIOBE Programming Community Index

Source: www.tiobe.com

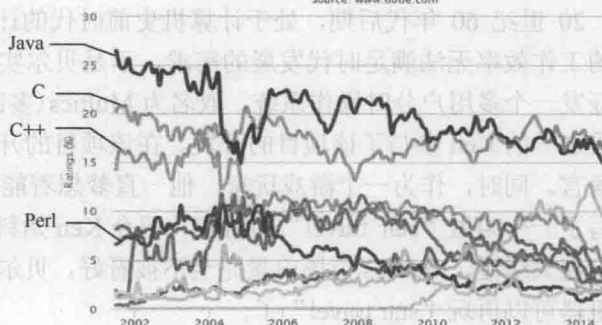


图 1.3 TIOBE 主流编程语言走势图

(1) C 语言至今仍然流行的原因是它无处不在，靠近硬件，被用于开发其他语言和操作系统；

(2) 通过学习 C 语言，可从系统级上更好地理解程序的工作原理；

(3) 大学应该继续教 C 语言，它是一种非常重要的基础性语言，要求完全理解整个技术栈；

(4) 学会 C 语言后学生将能更深入地理解计算机。

浅显地来讲，对于一直要使用 C 语言的人而言，真正掌握 C 语言，可以帮助他在编程道路上攻城拔寨，所向披靡；对于要编程又不用 C 语言的人而言，可以通过 C 语言窥探计算机底层原理，掌握最流行的语法形式，了解基本的程序设计思维；对于不编程的人而言，能通过 C 语言更好地了解计算机，了解计算思维，并且通过全国计算机等级二级考试；对所有人而言，C 语言会让你了解它并爱上它。

现在，有想学习 C 语言的冲动了吗？那么，就从追溯 C 语言的历史开始，我们一起进入编程这个神奇的魔法学校，学习 C 语言这种最有力的魔咒，成为最强的魔法师！

## 1.2 C 语言出现的历史背景

### 1.2.1 C 语言的诞生

谈及 C 语言的诞生，不能不介绍这两位公认的重量级编程高手，“骨灰级”游戏玩家，黑客的鼻祖：Ken Thompson (简称 Ken，图 1.4) 和 Dennis M. Ritchie (简称 DMR，图 1.5)。



图 1.4 Ken Thompson



图 1.5 Dennis M. Ritchie

20 世纪 60 年代后期，处于计算机史前时代的计算机系统还是批处理一统天下，但其极低的工作效率无法满足时代发展的需求，于是贝尔实验室、麻省理工学院、通用电气公司联合开发一个多用户分时操作系统，取名为 Multics (多路信息计算系统)，当时作为贝尔实验室成员之一的 Ken 参与了该项目的开发。在该项目的开发过程中，Ken 创造出了名为 Bon 的编程语言。同时，作为一个游戏玩家，他一直梦想着能做游太空，于是基于 Multics，Ken 自己编写了一个叫做“star travel”的游戏。但令 Ken 始料未及的是，由于开发周期长，成本高，而且庞大缓慢，Multics 市场前景完全不被看好，贝尔实验室撤出了该项目。这意味着 Ken 没有机器可以再玩“star travel”了。

为了满足玩游戏的需要，Ken 决定自给自足，在一台废弃已久的老式 PDP-7 小型机上重写他的游戏，因为这台机器是免费的。但最直接的问题就是：这台 PDP-7 上没有操作系统！逢山开路，遇水搭桥，Ken 勇往直前，用一个月的时间，利用 PDP-7 上的汇编语言，开发出了一个全新的操作系统，将其命名为 UNiplexed Information and Computing System，缩写为 UNICS。用汇编编写操作系统不方便，Ken 便开发出一种简洁明了且接近于硬件语言的新语言，即 C 语言的前身——B 语言，并用其改写了 UNICS，即产生了第一版的 UNIX。

UNIX 的出现开始并不被看好，但却引起了贝尔实验室另一位同事的注意，这就是同样酷爱“star travel”的 Dennis M. Ritchie，于是 DMR 主动加入共同完善这个系统。自 1972 年 UNIX 成功移植到当时最先进的大型机 PDP-2 上，并于 1973 年在 IBM 举办的研讨会上展示后，其简洁、稳定与高效的优秀性能表现立刻引起了轰动，得到大家的喜爱，需求量与日俱增。

UNIX 的原码中不少是用汇编完成的，不具备良好的移植性。为了可以移植到各种不同的硬件系统，Ken 与 DMR 决定进一步改写 UNIX。1973 年，DMR 在 B 语言的基础上开发出了 C 语言。C 语言灵活，高效性，与硬件无关，并且不失其简洁性，正是 UNIX 移植所需要的法宝，于是旧版的 UNIX 与 C 语言完美结合在一起产生了新的可移植的 UNIX 系统。UNIX

和 C 语言在发展过程中相辅相成，随着 UNIX 的广泛使用，C 语言也成为了当时最受欢迎的编程语言一直延续至今。

由于 UNIX 与 C 语言的深远影响，1983 年美国计算机协会将当年的图灵奖破例颁给了软件工程师 Ken 与 DMR，并在当年还决定新设立一个奖项——软件系统奖，以奖励那些优秀的软件开发者，当然首个软件系统奖也是非他们两人莫属了。1999 年两人因发展 C 语言和 UNIX 操作系统的杰出贡献一起获得了美国国家技术奖章。

## 1.2.2 C 语言标准的演变

### 1. K & RC

为了使 UNIX 操作系统推广，1977 年 DMR 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》，即是著名的 PCC (Portable C Compiler, 可移植 C 编译器)。1978 年，Brain W. Kernighan (简称 BWK) 和 DMR (两人合称 K&R) 合著 *The C Programming Language*。该书被誉为 C 语言的“圣经”，其中介绍的 C 语言被称为“标准 C”，人们通常称这个版本的 C 语言为 K & RC，是后来广泛使用的 C 语言的基础。即使在后来 ANSI C 标准被提出的许多年后，K & RC 仍然是许多编译器的最低标准要求，许多老旧的编译系统仍然运行 K & RC 的标准。

### 2. ANSI C/C 89 标准

20 世纪 70~80 年代，C 语言被广泛应用，从大型主机到小型微机，也衍生了 C 语言的很多不同版本。1983 年美国国家标准局 (American National Standards Institute, ANSI) 成立了一个委员会，来制定 C 语言标准。

1989 年 C 语言标准被批准，被称为 ANSI X3.159—1989 “Programming Language C”，简称 ANSI C 或 C89 标准。

1990 年，国际标准化组织 (International Organization for Standards, ISO) 接受了 1989 年的 ANSI C 为 ISO C 的标准，命名为 ISO/IEC 9899:1990，简称 C90 标准。不过习惯上仍然叫 C89 标准。

1994 年，ISO 修订了 C 语言的标准。

1995 年，ISO 对 C90 做了一些修订，即“1995 基准增补 1 (ISO/IEC/9899/AMD1:1995)”。

在 ANSI 标准化后，尽管 C 语言继续在改进，但 C 语言的标准在一段相当的时间内都保持不变。

### 3. C99 标准

在 ANSI 的标准确立后，C 语言的规范在一段时间内没有大的变动，然而 C++ 在自己的标准化创建过程中继续发展壮大。《标准修正案一》在 1994 年为 C 语言创建了一个新标准，但是只修正了一些 C89 标准中的细节和增加更多更广的国际字符集支持。不过，这个标准引出了 1999 年 ISO 9899:1999 的发表，它通常被称为 C99。C99 被 ANSI 于 2000 年 3 月采用。

但是各个公司对 C99 的支持所表现出来的兴趣不同。当 GCC 和其他一些商业编译器支持 C99 的大部分特性的时候，微软和 Borland 却似乎对此不感兴趣。

### 4. C11 标准

2011 年 12 月 8 日，ISO 正式发布了新的 C 语言的新标准 C11，之前被称为 C1X，官方名称为 ISO/IEC 9899:2011。新的标准提高了对 C++ 的兼容性，并增加了一些新的特性。

## 1.3 C 语言的特点

### 1.3.1 基本特性

(1) 低级的高级语言。从 C 语言的处理方式来看,它属于高级语言,但从性能和控制来看,它将高级语言的基本结构和语句与低级语言的实用性相结合,相比较其他的编程语言(像 C++, Java)而言,更接近低级语言。从总体上来说,可将 C 语言视为介于汇编语言和高级语言之间的一种语言。而低级的编程语言可以更好地了解计算机。

(2) 结构式语言。结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,它本身只提供必要的语言特性。这些函数不是作为 C 语言核心出现,可方便的调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(3) 功能齐全。具有各种各样的数据类型,并引入了指针概念,可使程序效率更高。而且计算功能、逻辑判断功能也比较强大,可以实现决策目的的游戏。

(4) C 语言兼容性、可移植性强,应用广。C 语言对编写所需的硬件环境兼容性优于其他高级语言,适用于多种操作系统,如 Windows、DOS、UNIX 等等;也适用于多种机型。在 ANSI C 制订后,只要特定平台上的编译器完整实现了 C89 标准,而且代码没有使用某些特殊的扩展(GCC 以及微软都有自己的编译器特定扩展),那么代码一定可以编译通过,再实现一下操作系统相关的函数库,C 语言的移植就是很简单的事情,“一次编写,处处编译”。

C 语言也是一个比较少见的应用领域极为广泛的语言。很多大型软件都是用 C 语言编写的,比如编写操作系统。C 语言可以编写服务器端软件,如 Apache、Nginx,或者编写 GUI 程序,如 GTK。大多数程序语言的第一版是通过 C 语言实现的,借助前面提到的“一次编写处处编译”,也最大程度保证了这些程序语言的可移植性。

(5) 指针。可以直接进行靠近硬件的操作,但是 C 的指针操作不作保护,也给它带来了许多不安全的因素。C++在这方面做了改进,在保留了指针操作的同时又增强了安全性,受到了一些用户的支持,由于这些改进增加了语言的复杂度,也为另一部分用户所诟病。Java 则吸取了 C++的教训,取消了指针操作,也取消了 C++改进中一些备受争议的地方,在安全性和适合性方面均取得良好的效果,但其本身解释在虚拟机中运行,运行效率低于 C++/C。

(6) C 语言文件由数据序列组成,可以构成二进制文件或文本文件。

### 1.3.2 优点

(1) 简洁紧凑、灵活方便。C 语言一共只有 32 个关键字,9 种控制语句,程序书写形式自由,区分大小写。把高级语言的基本结构和语句与低级语言的实用性结合起来。

C 语言相比 C++的优点之一就是最小原则,不会在编译时自作主张地自动产生一些额外代码。对于编写操作系统这类性能优化要求较高的系统时,这些多余的代码是不应该出现的。C 语言的简洁设计哲学让其具有优秀的可移植性、便携性,也使得很多嵌入式系统依然使用 C 语言作为主要编程语言。



(2)运算符丰富。C语言的运算符包含的范围很广泛，共有34种运算符。C语言把括号、赋值、强制类型转换等都作为运算符处理。从而使C语言的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(3)数据类型丰富。C语言的数据类型有：整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构的运算。并引入了指针概念，使程序效率更高。

(4)表达方式灵活实用。C语言提供多种运算符和表达式值的方法，对问题的表达可通过多种途径获得，其程序设计更主动、灵活。它语法限制不太严格，程序设计自由度大，如对整型量与字符型数据及逻辑型数据可以通用等。

(5)允许直接访问物理地址，对硬件进行操作。由于C语言允许直接访问物理地址，可以直接对硬件进行操作，因此它既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位(bit)、字节和地址进行操作，而这三者是计算机最基本的工作单元，可用来写系统软件。

(6)生成目标代码质量高，程序执行效率高。C语言描述问题比汇编语言迅速，工作量小、可读性好，易于调试、修改和移植，而代码质量与汇编语言相当。C语言一般只比汇编程序生成的目标代码效率低10%~20%。

(7)可移植性好。C语言在不同机器上的C编译程序，86%的代码是公共的，所以C语言的编译程序便于移植。在一个环境上用C语言编写的程序，不改动或稍加改动，就可移植到另一个完全不同的环境中运行。

### 1.3.3 缺点

(1)C语言的缺点主要表现在数据的封装性上，这一点使得C在数据的安全性上有很大缺陷，这也是C和C++的一大区别。

(2)C语言的语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不作检查等。

(3)从应用的角度，C语言比其他高级语言较难掌握。也就是说，对用C语言的人，要求对程序设计更熟练一些。

(4)在Web开发领域，C语言的应用相对较少，这也是一种取舍的结果。Web开发需要使用PHP、Ruby、Python这样的动态语言，可以快速上线快速修改，最大程度满足用户变化的需求，这也是C语言的弱项。如果把程序语言的应用领域从硬件到管理软件、Web程序做一个从下到上的粗略排列，C语言适合领域是比较底层靠近硬件的部分，而新兴语言比较偏重于高层管理或者Web开发这种相对贴近最终用户的领域。比较流行的混合开发模式是使用C语言编写底层高性能部分代码或后台服务器代码，而使用动态语言如PHP做前端开发，充分发挥它们各自的优势力量。

提到C语言的缺点，常常是它缺少这种或者那种特性，比如有人建议加入GC，有人建议加入并行或者并发支持，有人提到没有一个比较完整的类似C++的异常策略。这些特性有的可以通过引入第三方库来实现，但C语言的设计哲学其实决定了它不会像C++那样“非常强大”。即使引入了某些人期望的特性，依然会有某些人喜欢某些人不喜欢。现在的功能对于C语言应用领域来说已经够用，其他特性可以通过特定程序语言实现，并且通过CABI与C语言编写的程序进行交互。任何一个工匠都不可能只使用一个工具完成他的工作，不同工具结合起来才能更快更好地完成任务。