

高等学校通识教育系列教材

C/C++程序设计

张志强 周克兰 郑红兴 编著



清华大学出版社

高等学校通识教育系列教材



C/C++程序设计

张志强 周克兰 郑红兴 编著



清华大学出版社
北京

内 容 简 介

本书全面、系统地介绍 C 及 C++ 程序设计语言的基本概念、语法和编程方法。全书共分为 14 章, 都从实际应用出发, 蕴含了作者丰富的教学经验和编程心得。第 1 章通过一个简单 C 语言程序的编写、编译和运行介绍程序设计的基本概念; 第 2 章从计算机内数据存储的角度介绍 C 语言程序中数据类型的概念及使用方法; 第 3 章介绍 C 语言提供的各种运算功能; 第 4 章讲述结构化程序设计的方法; 第 5 章讲述使用数组处理大量数据的方法; 第 6 章讲述各种自定义数据类型的使用方法; 第 7 章讲述使用函数进行模块化程序设计的方法, 并重点讲述指针在函数参数中的作用; 第 8 章讲述使用指针操作动态内存的方法及链表基本应用方法; 第 9 章讲述在 C 语言中处理文件的基本方法; 第 10 章介绍 C 语言的预处理命令及简单应用; 第 11 章讲述 C++ 语言相对于 C 语言的部分扩展功能; 第 12 章讲述使用 C++ 语言进行面向对象程序设计的基本方法; 第 13 章讲述使用 MFC 应用程序框架编写 Windows 应用程序的基本方法; 第 14 章参考全国计算机等级考试二级考试大纲, 介绍在软件开发过程中常用的编程技术。

全书内容由浅入深, 面向实际应用, 例题经典、丰富, 将指针的应用融合到全书各章。本书结构新颖、内容紧凑、通俗易懂, 是从零起步学习 C 及 C++ 语言的合适教材。本书既可以作为普通本科院校、普通高等专科学校的计算机教材, 也可作为计算机培训和计算机等级考试辅导的教学用书。

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/C++ 程序设计/张志强等编著. —北京: 清华大学出版社, 2019 (2019.9 重印)

(高等学校通识教育系列教材)

ISBN 978-7-302-52435-9

I. ①C… II. ①张… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 039287 号

责任编辑: 刘向威

封面设计: 文 静

责任校对: 梁 毅

责任印制: 李红英

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

印 装 者: 北京九州迅驰传媒文化有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 29.25 字 数: 712 千字

版 次: 2019 年 4 月第 1 版 印 次: 2019 年 9 月第 2 次印刷

印 数: 2501~2700

定 价: 69.00 元

产品编号: 082997-01

前 言

从诞生之日起，C 及 C++ 语言就作为应用最为广泛的程序设计语言长盛不衰，从家用电器中的单片机到企业生产设备中的工业控制系统，从汽车中的车载计算机到轮船、宇宙飞船中的控制系统，从家用计算机到巨型机，可以说只要有计算机的地方，C 及 C++ 语言都是最重要的两种程序设计语言。由于 C++ 语言是 C 语言的超集，本书前半部分完整介绍 C 语言程序设计，后半部分介绍 C++ 语言对于 C 语言的扩充及常用编程技术。读者也可以暂时跳过标记*部分内容的学习，即使只学习本书的前半部分内容，也可以完整学习 C 语言程序设计方法，以后再学习 C++ 相对于 C 语言的扩充部分亦可事半功倍。

本书主要作者为一直工作于教学一线的大学教师，承担“C 语言程序设计”与“C++ 语言程序设计”课程的教学任务十余年，有着丰富的教学经验，同时长期从事 C 及 C++ 语言软件开发工作，有几十万行 C/C++ 程序代码的开发经验，开发的软件多次获得省级、市级奖励。在教学实践中，作者感受最深的就是学生普遍反映 C 及 C++ 语言难学难懂，尤其是指针千变万化，难以捉摸，而事实上，只要遵照一定的学习规律，C 及 C++ 语言并不难掌握。

部分 C 及 C++ 语言教材过细的内容组织让学生迷失了方向。看着满篇的烦琐内容，读者根本不明白该学什么，更不明白 C 与 C++ 的重点是什么，唯一的感受就是 C 和 C++ 语言难而烦。本书力求做到去繁就简，以弄懂基本的、主要的、核心的内容为重点，紧紧围绕软件开发需求最核心的内容进行组织。

基于作者长期从事软件开发的经验，本书特别强调实践能力的培养。上机练习是学习程序设计的最好方法，本书内容安排在读者学完第 1 章后就可以开始。书中每章内容都包含大量示例，课后练习也以编程实践为主，课后练习的内容由浅入深，如果能做完，本章内容即可完全掌握。

本书第 1 章阐述计算机及程序的基本工作原理，C 语言程序是怎样运行的，并通过一个完整的 C 程序例子介绍 C 程序的各部分组成、功能及 C 语言程序编辑、编译、运行的方法。通过本章的学习，使读者能够对 C 语言程序及程序设计的过程和方法有一个基本的认识。

第 2 章用了大量的篇幅介绍计算机中数据的存储方式，从而引入了整数、浮点数、指针等数据类型的概念，并使读者理解这些数据类型的作用和意义。在本章最后介绍 C 语言中输入、输出的基本方法，使读者马上可以通过这些输入、输出功能，对刚学到的各种不同类型数据进行比较和分析。

第 3 章讲解 C 语言中可以使用的各种运算符，并通过大量的示例来展示这些运算符的功能。本章还详细讲解了指针的各种运算方法，为以后指针的使用做好准备。

通过前面 3 章的学习，读者已经掌握了 C 语言中各种基本数据类型的处理方法。

第 4 章的内容通过讲解顺序结构、选择结构、循环结构这三种程序结构，使读者可以编写出具有一定实用功能的程序。本章的难点是循环，尤其是多重循环，这也是很多学生在学习 C 语言过程中开始掉队的地方，解决这个难点的唯一方法就是练习，反复地编程练习。

第 5 章讲述数组，使 C 语言程序可以处理大量的数据。数组的处理离不开循环，所以本章的内容还包括了第 4 章循环内容的强化。另外，由于数组的元素在内存中是连续存储的，这在本书中第一次给了指针大显身手的空间，完成本章内容的学习，C 语言已经入门一半了。

第 6 章讲述包括结构体在内的 C 语言各种自定义数据类型的使用方法。本章以概念性内容居多，虽然烦琐但并不难以掌握，书中通过各种示例对它们的定义、方法和用法一一进行了展示，是经过第 5 章艰苦学习过程后的一次小的休整。

第 7 章讲述 C 语言中函数的使用方法及模块化程序设计的基本思想，通过将一个复杂程序划分成若干函数来实现，从而降低了程序的编写难度。在函数的调用过程中，指针作为函数参数可以起到双向传值的作用，这些都是本章的重点和难点。

第 8 章讲述动态内存的使用方法，操作内存离不开指针，本章首先讲述如何获取动态内存，然后讲述通过链表来组织、使用动态内存的方法。链表是结构体、指针的结合，由于操作的复杂性使得使用函数成为必然，所以说本章内容是前 7 章内容的综合。完成本章内容的学习，C 语言的掌握可以算入门了。

第 9 章讲述在 C 语言中操作文件的方法，主要是一些文件操作函数的应用。

第 10 章讲述在 C 语言中一些编译预处理命令的使用方法，在本章结尾处介绍了在组织多文件的 C 语言源程序时条件编译的应用，为读者以后编写大型 C 语言程序提供了方便。

第 11 章讲述 C++ 语言相对于 C 语言的部分扩展功能。通过本章的学习，读者可以了解 C 语言与 C++ 语言的区别，初步了解使用 C++ 语言进行程序设计的方法。

第 12 章讲述使用 C++ 语言进行面向对象程序设计的基本方法。通过本章的学习，读者可以了解面向对象程序设计的基本原理及使用 C++ 语言进行面向对象程序设计的基本方法。完成本章内容的学习，C++ 语言的掌握可以算入门了。

第 13 章讲述使用 MFC 应用程序框架编写 Windows 应用程序的基本方法。标准的 C 及 C++ 语言都不涉及具体操作系统下的软件开发功能，本章介绍使用微软公司提供的 MFC 应用程序框架开发 Windows 应用程序的方法，通过本章的学习，读者可以掌握使用 C++ 语言开发 Windows 系统下图形界面应用程序的方法和技术。

第 14 章介绍一些常用的编程技术。该部分内容全面覆盖了全国计算机等级考试二级考试大纲中公共知识部分的内容。通过本章及前 10 章内容的学习，读者可以达到全国计算机等级考试二级 C 语言考试的要求。通过本章及前 12 章内容的学习，读者可以达到全国计算机等级考试二级 C++ 语言考试的要求。通过全书的学习，读者可以达到入门级程序员水平。

本书在编写过程中参考了许多同行的著作，作者在此一并表示感谢。

本书的编写得到了江苏省教育改革课题（2017JSJG532）的资助。感谢为本书提供直接或间接帮助的每一位朋友，你们的帮助和鼓励促成了本书的顺利完成。如果您能够愉快地读完本书，并告知身边的朋友，原来 C 及 C++ 语言并不难学，那么作者编写本书的目的就达到了。

尽管作者尽了最大努力，但是由于时间关系及作者学识所限，书中难免存在缺点和错误，恳请各位读者批评指正，以便再版时修订。

编者
2019 年 1 月

目 录

第 1 章 C 语言导论	1
1.1 C 语言概述	1
1.1.1 C 语言的功能	1
1.1.2 C 语言的起源	2
1.1.3 C 语言的学习阶段与学习方法	3
1.2 第一个 C 语言程序	4
1.2.1 程序代码	4
1.2.2 空白和注释	5
1.2.3 预处理指令	5
1.2.4 main 函数	6
1.2.5 程序输出	6
1.3 C 语言程序的运行	7
1.3.1 程序的编译	7
1.3.2 程序的运行和调试	8
1.4 习题	9
1.5 阅读材料——UNIX 和 C 的故事*	10
第 2 章 数据	14
2.1 基本数据类型	14
2.1.1 整型数据	14
2.1.2 浮点型数据	17
2.1.3 指针型数据	17
2.2 常量	18
2.2.1 字面常量	18
2.2.2 符号常量	22
2.3 变量	24
2.3.1 标识符	24
2.3.2 变量的定义	25
2.3.3 变量的初始化	26
2.3.4 变量的使用	27
2.4 输出与输入	28

2.4.1	基本输出	28
2.4.2	基本输入	32
2.5	习题	34
2.6	阅读材料——二进制与计算机*	36
2.6.1	二进制起源	36
2.6.2	计算机与二进制	36
2.6.3	进制转换	37
2.6.4	计算机中的补码	38
第3章	运算与表达式	40
3.1	算术运算	40
3.1.1	基本算术运算符	40
3.1.2	优先级与结合性	42
3.1.3	数据类型转换	43
3.1.4	自增、自减运算	45
3.2	关系运算	46
3.3	逻辑运算	46
3.4	位运算	48
3.4.1	按位逻辑运算	48
3.4.2	移位运算*	50
3.4.3	程序例子	52
3.5	指针运算	52
3.5.1	取地址运算	52
3.5.2	操作指针变量	55
3.5.3	移动指针	57
3.5.4	比较指针	58
3.6	其他运算	58
3.6.1	sizeof 运算	58
3.6.2	逗号运算	60
3.6.3	条件运算	60
3.7	赋值运算	61
3.7.1	赋值运算符和赋值表达式	61
3.7.2	复合赋值运算	62
3.8	习题	63
第4章	程序控制结构	66
4.1	程序语句	66
4.2	顺序结构	67
4.3	选择结构	69

4.3.1	if 语句	69
4.3.2	if 嵌套	72
4.3.3	switch 语句	78
4.4	循环结构	83
4.4.1	while 循环	83
4.4.2	do-while 循环	86
4.4.3	for 循环	89
4.4.4	其他控制语句	90
4.4.5	循环控制嵌套	92
4.5	习题	96
第 5 章	数组	98
5.1	一维数组	98
5.1.1	一维数组的定义	98
5.1.2	一维数组元素的引用	100
5.1.3	一维数组的初始化	102
5.1.4	程序举例	104
5.2	多维数组	105
5.2.1	多维数组的定义	106
5.2.2	多维数组元素的引用	109
5.2.3	多维数组的初始化	112
5.2.4	程序举例	113
5.3	字符数组与字符串	115
5.3.1	字符数组与字符串的关系	115
5.3.2	字符串的输入输出	116
5.3.3	字符串处理函数	119
5.3.4	程序举例	122
5.4	指针数组*	126
5.4.1	指针数组的定义与应用	126
5.4.2	指向指针的指针	130
5.5	习题	132
5.6	阅读材料——字符集、字符编码、C 语言宽字符间的关系*	132
第 6 章	其他数据类型	137
6.1	结构体类型的定义	137
6.2	结构体类型变量	138
6.2.1	结构体变量的定义	138
6.2.2	结构体变量的引用	140
6.2.3	结构体变量的初始化	144

6.3	结构体类型数组	145
6.3.1	结构体数组的定义	145
6.3.2	结构体数组的初始化	146
6.3.3	结构体数组的引用	147
6.4	位段类型*	149
6.4.1	位段成员的定义	150
6.4.2	位段成员的使用	151
6.5	共用体类型*	154
6.5.1	共用体类型的定义	154
6.5.2	共用体变量的定义	155
6.5.3	共用体变量的引用	157
6.5.4	共用体变量的初始化	160
6.5.5	共用体变量的应用	161
6.6	枚举类型	163
6.6.1	枚举类型的定义	163
6.6.2	枚举类型变量的定义与引用	164
6.7	typedef 自定义类型	166
6.7.1	typedef 定义类型	166
6.7.2	typedef 应用举例*	167
6.8	习题	169
第7章	函数	171
7.1	函数的定义和调用	171
7.1.1	函数概述	171
7.1.2	函数的定义	172
7.1.3	函数的调用	172
7.2	函数的返回值、参数及函数声明	173
7.2.1	函数的返回值	173
7.2.2	函数的参数	177
7.2.3	函数的声明	180
7.3	函数的嵌套和递归调用	182
7.3.1	函数的嵌套调用	182
7.3.2	函数的递归调用	185
7.4	函数与指针	189
7.4.1	指针变量作为函数参数	189
7.4.2	数组与函数	192
7.4.3	返回指针值的函数	195
7.4.4	指向函数的指针*	196

7.5	作用域	199
7.5.1	局部作用域	199
7.5.2	全局作用域	202
7.5.3	多文件下的全局作用域	204
7.6	变量的存储类别	208
7.6.1	动态存储方式	208
7.6.2	静态存储方式	211
7.7	习题	212
第 8 章	内存的使用	214
8.1	动态使用内存	214
8.1.1	分配内存	214
8.1.2	释放内存	215
8.1.3	应用举例	215
8.2	链表	219
8.2.1	链表概述	219
8.2.2	创建链表	223
8.2.3	释放链表	233
8.2.4	链表操作*	236
8.3	习题	239
第 9 章	文件	240
9.1	文件概述	240
9.1.1	数据文件	240
9.1.2	文件的读写	240
9.1.3	文件指针	241
9.1.4	文件操作的步骤	241
9.2	文件的打开与关闭	241
9.2.1	打开文件	241
9.2.2	关闭文件	243
9.3	文件的读写	243
9.3.1	字符读写文件	244
9.3.2	字符串读写	246
9.3.3	格式化读写文件	248
9.3.4	非格式化读写文件	250
9.4	文件的随机读写*	251
9.4.1	文件定位	251
9.4.2	应用举例	252
9.5	习题	254

第 10 章	编译预处理	255
10.1	宏定义.....	255
10.1.1	无参宏定义.....	255
10.1.2	带参宏定义.....	257
10.2	文件包含.....	260
10.3	条件编译*.....	261
10.3.1	条件编译的形式.....	261
10.3.2	条件编译与多文件组织.....	263
10.4	习题.....	264
第 11 章	C++对 C 的扩充	265
11.1	C++语言概述.....	265
11.1.1	C++的起源.....	265
11.1.2	C++的面向对象程序设计.....	265
11.1.3	C++的泛型程序设计.....	266
11.1.4	C++的过程化程序设计.....	267
11.2	C++的数据类型.....	268
11.2.1	常量.....	268
11.2.2	布尔.....	269
11.2.3	引用.....	270
11.2.4	C++的类型检查.....	271
11.3	C++的运算符.....	273
11.3.1	作用域运算符.....	273
11.3.2	new 运算符.....	274
11.3.3	delete 运算符.....	275
11.4	C++的名字空间.....	276
11.5	C++的输入、输出.....	278
11.5.1	基本输入、输出.....	278
11.5.2	文件输入、输出*.....	278
11.6	C++的函数与模板.....	280
11.6.1	内联函数.....	280
11.6.2	函数的重载.....	281
11.6.3	带默认参数的函数.....	282
11.6.4	函数模板*.....	283
11.7	C++的异常处理*.....	284
11.7.1	异常的概念.....	284
11.7.2	C++异常处理的方法.....	285
11.7.3	C++标准异常.....	286

11.8	习题	287
第 12 章	基于 C++ 的面向对象编程	288
12.1	类和对象	288
12.1.1	概述	288
12.1.2	类的定义	289
12.1.3	类和对象的特性	291
12.1.4	构造函数	292
12.1.5	析构函数	295
12.1.6	静态成员*	296
12.1.7	类的友元*	297
12.1.8	const 对象*	298
12.1.9	指向类成员的指针*	301
12.1.10	运算符重载*	302
12.2	类的继承与派生	304
12.2.1	继承与派生的概念	304
12.2.2	继承与派生的一般形式	305
12.2.3	派生类的构造函数与析构函数	307
12.2.4	继承与派生的访问权限*	308
12.3	类的多态性	310
12.3.1	多态性的概念	310
12.3.2	虚函数与动态多态	311
12.3.3	虚类(抽象类)*	314
12.3.4	虚基类*	315
12.4	类模板与泛型编程*	316
12.4.1	类模板	316
12.4.2	类模板的继承	318
12.4.3	泛型编程	318
12.5	习题	321
第 13 章	基于 MFC 的 Windows 编程	323
13.1	MFC 基础	323
13.1.1	概述	323
13.1.2	创建 MFC 框架程序	325
13.1.3	Windows 程序的资源	328
13.1.4	MFC 对话框程序	333
13.2	MFC 控件	335
13.2.1	按钮控件	336
13.2.2	静态控件	338

13.2.3	编辑控件	340
13.3	MFC 绘图	342
13.3.1	基本概念	342
13.3.2	CDC 类与绘图	343
13.3.3	GDI 类与绘图属性	346
13.3.4	修改控件的字体*	351
13.4	常用对话框	353
13.4.1	消息对话框	353
13.4.2	文件对话框*	354
13.5	常用消息	356
13.5.1	鼠标消息	356
13.5.2	键盘消息*	358
13.5.3	定时器消息*	361
13.5.4	关闭窗口消息*	363
13.6	习题	364
第 14 章	编程技术基础*	367
14.1	基本数据结构与算法	367
14.1.1	基本概念	367
14.1.2	线性表及其顺序存储结构	372
14.1.3	栈和队列	375
14.1.4	链表	378
14.1.5	树与二叉树	382
14.1.6	常见查找算法	386
14.1.7	常见排序算法	387
14.2	程序设计基础	392
14.2.1	程序设计方法与风格	392
14.2.2	结构化程序设计	393
14.2.3	面向对象的程序设计方法	395
14.3	软件工程基础	398
14.3.1	软件工程基本概念	398
14.3.2	结构化分析方法	403
14.3.3	结构化设计方法	405
14.3.4	软件测试的方法	409
14.3.5	程序的调试	411
14.4	数据库设计基础	411
14.4.1	数据库的基本概念	411
14.4.2	数据模型	412
14.4.3	关系代数运算	419

14.4.4 数据库设计方法和步骤	424
14.5 习题	430
附录 A 常用字符与 ASCII 码对照表	436
附录 B 运算符和结合性	437
附录 C 常用标准库函数	439
附录 D 几个常用 MFC 类	446
附录 E Windows 虚键码表	450
参考文献	452

1.1 C 语言概述

1.1.1 C 语言的功能

计算机的应用已经深入到社会各个领域，成为人们日常工作、生活、学习的必备工具。计算机是一种具有存储程序、执行程序能力的电子设备，计算机所有能力都是通过执行程序来实现的。程序就是人们把需要做的工作写成一定形式的指令序列，并把它存储在计算机内部存储器中，当人们给出命令之后，计算机就按照指令的执行顺序自动进行相应操作，从而完成相应的工作。人们把这种可以连续执行的一条条指令的序列称为“程序”。编写程序的过程就称为“程序设计”。为了使计算机能够正确识别和执行，不能随意编写指令序列，必须有一定的规则。这些规则包含了一系列文法和语法的要求，按照这些规则编写的程序能够被计算机理解执行，所以它是人和计算机之间的交流语言。这种语言类似于人与人之间交流的语言，虽然没有人类语言那么复杂，但逻辑上要求更加严格，符合这些规则的“语言”也被称为“程序设计语言”。

机器语言，也称作二进制代码语言，是计算机可以直接识别的语言。每台机器的指令，其格式和代码所代表的含义都是硬性规定的，如某种计算机的指令为 1011011000000000，它表示让计算机进行一次加法操作；而指令 1011010100000000 则表示进行一次减法操作。它们的前八位表示操作码，后八位表示地址码。因为硬件设计不同，机器语言对不同型号的计算机来说一般是不同的。用机器语言编程，就是从使用的 CPU 指令系统中挑选合适的指令，组成一个指令系列的过程。

由于机器语言与人们日常生活使用的语言差距过大，而且大量的规则都和具体的计算机硬件设计与实现相关，所以使用机器语言编写程序难度很大。为了降低编写程序的难度，人们发明了一些更加接近人类日常语言的程序设计语言，但这些语言编写的程序不能被计算机直接识别、执行，必须翻译成“机器语言程序”才能被计算机执行。

根据不同程序设计语言与人类语言的接近程度，可把这些语言分为高级语言、中级语言、低级语言。低级语言最接近机器语言，学习和使用难度都比较大；高级语言最接近人类语言，学习和使用难度相对低级语言要容易得多，应用最为广泛。目前常见的高级语言有 C、Java、C++、C#、Python、PHP 等。由程序设计语言编写的程序称为“源程序”，高级语言编写的程序不能被计算机硬件直接识别、执行，高级语言程序编译（compile）成机器语言程序的过程如图 1-1 所示。

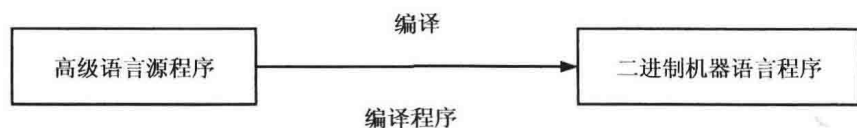


图 1-1 程序编译、链接过程示意图

几十年来，人们发明了很多种计算机程序设计语言，目前还不断有新的程序设计语言被发明出来，这些语言往往具有不同的特点。C 语言是目前应用最为广泛的高级程序设计语言，它是一种通用的高级程序设计语言，可以用来完成各种类型的应用软件设计。C 语言的通用性和无限制性使得它对于程序设计者来说都显得更加方便、有效。从微型计算机到小型机、中型机、大型机、巨型机，都离不开 C 语言编写的程序；家用冰箱、电视、洗衣机、空调、手机等，大部分都依赖内部运行的 C 语言编写的程序有效工作；现代化的智能车床、工业控制设备、汽车、火箭、宇宙飞船内部运行程序大部分也都是 C 语言编写的。可以说，有计算机的地方就有 C 语言编写的程序在运行。

1.1.2 C 语言的起源

在学习 C 语言之前，我们有必要了解一下 C 语言的发展历史。

计算机技术发展速度如此之快，以致让人目不暇接。作为计算机软件技术的基础，新的程序设计语言和新的操作系统也在不断涌现，这些新技术、新产品往往都是由国际著名企业或国家重要部门投入巨大的人力、物力所开发出来的。然而，提到目前最优秀、最有价值的程序设计语言和操作系统，却要回到四十年前，那就是强大的 C 语言和用 C 语言编写的 UNIX 操作系统。更加让人惊奇的是，它们竟然只是几个人基于个人兴趣所缔造的，花费的人力和物力代价更是少的可怜。

从历史发展的角度看，C 语言起源于 1968 年发表的 CPL 语言（combined programming language），它的许多重要思想来自于 Martin Richards 在 1969 年开发的 BCPL 语言，以及以 BCPL 语言为基础的 B 语言。丹尼斯·里奇（Dennis M. Ritchie，如图 1-2 所示）在 B 语言基础上，于 1972 年开发了 C 语言，并用 C 语言写成了第一个在 PDP-11 计算机上实现的 UNIX 操作系统（主要在贝尔实验室内部使用）。此后，又经过多次改进，直到 1975 年用 C 语言编写的 UNIX 操作系统第 6 版公之于世后，C 语言才举世瞩目。1977 年出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》，从而大大简化了把 C 语言编译程序移植到新环境所需做的工作，这本身也就使 UNIX 操作系统迅速在众多计算机上实现。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广。1978 年以后，C 语言先后移植到大、中、小、微型计算机上，它的应用领域已不再限于系统软件的开发，而成为当今最流行的程序设计语言。

以 1978 年发布的 UNIX 第 7 版 C 语言编译程序为基础，Brian W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的名著 *The C Programming Language*，书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，它被称为标准 C。

1983 年美国国家标准化协会（ANSI）根据 C 语言问世以来的各种版本，对 C 语言的发展和扩充制定了新的标准，称为 ANSI C。1989 年 ISO 根据 ANSI C 公布的 C 标准，即