应用型本科信息大类专业“十三五”规划教材

C语言程序设计

王海文 林 月 林 强 主编

華中科技大學出版社
http://www.hustp.com

应用型本科信息大类专业“十三五”规划教材

C语言程序设计

主　编　王海文　林　月　林　强
副主编　关　夺　王虹元　尚靖涵

华中科技大学出版社
http://www.hustp.com
中国 · 武汉

内 容 简 介

本书主要讲述数据类型、运算符与表达式、顺序结构、选择结构、循环结构、函数、数组、指针、结构体和共用体、文件等内容。

本书作为C语言程序设计的入门教材，一方面，内容由浅入深、通俗易懂，循序渐进地将各个知识点讲解清楚，引导学生顺利学习并掌握；另一方面，特别强调对学生程序设计能力的培养，主要是通过提供典型并具有一定趣味性的例题以及大量的习题，提高学生学习兴趣，进而达到刻苦专研、自觉学习的目的。

为了方便教学，本书还配有电子课件等教学资源包，任课教师和学生可以登录“我们爱读书”网(www.ibook4us.com)免费注册并浏览，或者发邮件至 hustpeiit@163.com 免费索取。

本书可以作为普通高等院校相关专业的教学用书，也可作为编程爱好者的自学参考书。

图书在版编目(CIP)数据

C语言程序设计/王海文，林月，林强主编. —武汉：华中科技大学出版社，2017.1
应用型本科信息大类专业“十三五”规划教材
ISBN 978-7-5680-2305-4

Ⅰ.①C… Ⅱ.①王… ②林… ③林… 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(2016)第 261101 号

C语言程序设计
C Yuyan Chengxu Sheji

王海文 林月 林强 主编

策划编辑：康 序
责任编辑：史永霞
封面设计：原色设计
责任监印：朱 玢
出版发行：华中科技大学出版社(中国·武汉) 电话：(027)81321913
武汉市东湖新技术开发区华工科技园 邮编：430223
录 排：武汉正风天下文化发展有限公司
印 刷：武汉市籍缘印刷厂
开 本：787mm×1092mm 1/16
印 张：18
字 数：496 千字
版 次：2017 年 1 月第 1 版第 1 次印刷
定 价：38.00 元

前言 PREFACE

C语言是一种结构化程序设计语言，具有编程方便、语句简练、功能强、移植性好等特点，是软件开发中最常用的计算机语言。它既具有高级语言的特点，又具有汇编语言的特点；既适用于系统软件的开发，也适合于应用软件的开发，而且其编写不依赖于计算机硬件。因此，C语言以其自身的特色和优势，在工业控制、嵌入式系统开发、单片机编程、系统底层开发等领域应用非常广泛。

大多数院校都将C语言作为入门的程序设计教学语言，同时将C语言程序设计作为计算机专业学生最重要的专业基础课程之一，因此对教师和学生都提出了较高的要求，并给予厚望。目前，国内已经出版的C语言教材非常多。作者在充分吸收这些教材的优点的同时，又将丰富的教学经验融入写作之中，以期达到教材充实、完整和便于学习的目的。

本书作为C语言程序设计入门教材，一方面注意到深入浅出、循序渐进地将各个知识点讲解清楚，引导学生顺利学习并掌握；另一方面，特别强调了对学生程序设计能力，主要是提供典型和具有提供一定趣味性的例题以及大量的习题，提高学生学习兴趣，进而达到刻苦钻研、自觉学习的目的。

本书中的所有案例程序代码均在Visual C++ 6.0环境下调试通过。语法知识点均符合C99标准。本书在教学使用过程中，可根据专业特点和课时安排选取教学内容。每章后面附有习题，可供学生课后练习。为了便于教学，本书有配套教材——《C语言程序设计实验指导与习题选解》。

需要特别指出的是：大部分同学都是第一次接触到C语言程序设计，而程序设计思想的建立和形成并非一日之功。因此，建议本课程除了理论学时之外，应该设置比较充足的实验学时，或者提供大量的课外上机时间。唯有这样，才能充分理解、巩固、强化和掌握课程的主要知识点，进而达到灵活自如地进行程序设计的目的。如果有条件，最好在学完本教材后安排一次课程设计，要求学生独立完成一个有一定规模的程序设计，这是一个重要的教学实践环节，能大大提高学生的独立编程能力。

本书由大连工业大学王海文、大连工业大学艺术与信息工程学院林月、林强担任主编，大连工业大学艺术与信息工程学院关夺、王虹元、尚靖函担任副主编。

本书内容共12章：王海文老师编写第5、6、7章，林月老师编写第1、2、3、4章，林强老师编写第8、9、10章，关夺老师编写第11章，王虹元老师编写第12章，尚靖函老师编写附录。黄婷婷、蔡国昱、王艺菲、王中鑫协助进行了资料的整理工作。

在编写本书的过程中，我们参考了兄弟院校的资料及其他相关教材，并得到许多同人的关心和帮助，在此谨致谢意。

为了方便教学，本书还配有电子课件等教学资源包，任课教师和学生可以登录“我们爱读书”网(www.ibook4us.com)免费注册并浏览，或者发邮件至hust-peiit@163.com免费索取。

限于篇幅及编者的业务水平，虽然我们付出了最大努力，但是书中难免存在不足甚至错误之处，敬请广大读者批评指正。

编者

2016年11月

目录

CONTENTS

第 1 章　C 语言程序设计基础 …… (1)
1.1　程序与程序设计语言 …… (1)
1.1.1　程序与指令 …… (1)
1.1.2　程序设计语言 …… (1)
1.1.3　高级语言程序的开发过程 …… (2)
1.2　C 语言概述 …… (4)
1.2.1　C 语言的产生与发展 …… (4)
1.2.2　C 语言的特点 …… (5)
1.2.3　C 语言的应用 …… (5)
1.3　C 语言开发程序 …… (6)
1.3.1　用 C 语言开发程序的过程 …… (6)
1.3.2　算法的概念和特征 …… (8)
1.3.3　结构化程序设计方法 …… (8)
1.3.4　算法的表示 …… (9)
1.3.5　C 语言程序的结构 …… (11)
1.4　C 语言程序的实现 …… (13)
1.4.1　C 语言程序的开发过程 …… (13)
1.4.2　VC++ 6.0 集成开发环境 …… (14)
1.4.3　C 语言运行环境 …… (19)
习题 …… (20)
第 2 章　数据类型、运算符和表达式 …… (23)
2.1　从数学上的“数”过渡到计算机中的“数” …… (23)
2.2　数据类型概述 …… (24)
2.3　常量和变量 …… (25)
2.3.1　常量 …… (25)
2.3.2　变量 …… (25)
2.3.3　变量名规则 …… (26)

2.3.4 变量的定义 …… (26)
2.4 整数类型 …… (27)
2.4.1 整型常量 …… (27)
2.4.2 整型变量 …… (27)
2.5 实数类型 …… (29)
2.5.1 实型常量 …… (29)
2.5.2 实型变量 …… (30)
2.6 字符类型 …… (30)
2.6.1 字符型常量 …… (31)
2.6.2 字符型变量 …… (32)
2.6.3 字符数据在内存中的存储形式及其使用 …… (32)
2.7 不同数据类型之间的转换 …… (33)
2.7.1 自动类型转换 …… (33)
2.7.2 强制类型转换 …… (34)
2.8 运算符与表达式 …… (35)
2.8.1 算术运算符与算术表达式 …… (35)
2.8.2 赋值运算符与赋值表达式 …… (37)
2.8.3 关系运算符与关系表达式 …… (38)
2.8.4 逻辑运算符与逻辑表达式 …… (39)
2.8.5 条件运算符与条件表达式 …… (41)
2.8.6 逗号运算符与逗号表达式 …… (42)
2.9 运算符与表达式的综合练习 …… (42)
2.9.1 正确的C语言表达式书写 …… (42)
2.9.2 复杂表达式的分析 …… (43)
习题 …… (45)
第3章 顺序程序设计 …… (48)
3.1 C语句 …… (48)
3.1.1 C语句概述 …… (48)
3.1.2 赋值语句 …… (49)
3.2 字符数据的输入/输出 …… (50)
3.2.1 数据输入/输出的概念及在C语言中的实现 …… (50)
3.2.2 字符的输出函数 putchar 函数 …… (50)
3.2.3 字符的输入函数 …… (51)
3.3 格式化输入与输出函数 …… (54)
3.3.1 格式化的输出函数 printf 函数 …… (54)
3.3.2 格式化的输入函数 scanf 函数 …… (58)
3.4 顺序结构程序设计举例 …… (62)
习题 …… (63)
第4章 选择结构 …… (65)

4.1 if 语句 …… (65)
4.1.1 单选择结构 …… (65)
4.1.2 if-else 语句 …… (68)
4.1.3 if-elseif-else 语句 …… (70)
4.1.4 嵌套 if 结构 …… (74)
4.2 switch 结构 …… (75)
4.3 多重 if 结构和 switch 结构的比较 …… (77)
4.4 应用举例 …… (79)
习题 …… (80)
第 5 章 循环结构 …… (87)
5.1 循环结构简介 …… (87)
5.2 while 语句 …… (87)
5.3 do-while 语句 …… (90)
5.4 for 语句 …… (92)
5.4.1 基本的 for 语句 …… (92)
5.4.2 各种特殊形式的 for 语句 …… (94)
5.5 三种循环语句的比较 …… (96)
5.6 break 语句 …… (97)
5.7 continue 语句 …… (98)
5.8 嵌套循环 …… (100)
5.9 应用举例 …… (103)
习题 …… (105)
第 6 章 函数 …… (111)
6.1 函数概述 …… (111)
6.2 函数的定义与调用 …… (113)
6.2.1 函数的定义 …… (113)
6.2.2 函数的调用、参数及传递方式 …… (114)
6.2.3 函数的返回值 …… (117)
6.2.4 函数声明的作用 …… (119)
6.2.5 main 函数中的参数 …… (120)
6.3 函数的嵌套调用与递归调用 …… (120)
6.3.1 函数的嵌套调用 …… (120)
6.3.2 函数的递归调用 …… (122)
6.4 变量的作用域与存储类型 …… (127)
6.4.1 变量的作用域 …… (127)
6.4.2 全局变量 …… (128)
6.4.3 变量的存储类型 …… (129)
6.5 常用系统函数 …… (135)
6.5.1 数学函数 …… (135)

6.5.2 输入输出函数 …… (136)
6.5.3 时间函数 …… (137)
6.5.4 随机数函数 …… (137)
习题 …… (138)
第 7 章 数组 …… (143)
7.1 数组的概念 …… (143)
7.2 数组的定义 …… (144)
7.2.1 一维数组 …… (144)
7.2.2 二维数组 …… (149)
7.3 数组作为函数的参数 …… (153)
7.3.1 用数组元素作函数实参 …… (153)
7.3.2 用数组名作函数参数 …… (154)
7.3.3 用多维数组名作函数参数 …… (156)
7.4 数组应用举例 …… (157)
7.5 字符串 …… (165)
7.5.1 字符串概念 …… (165)
7.5.2 字符串函数 …… (168)
7.5.3 字符串应用举例 …… (171)
习题 …… (174)
第 8 章 指针 …… (180)
8.1 指针的概念 …… (180)
8.2 指针变量 …… (180)
8.2.1 指针定义 …… (180)
8.2.2 指针运算符(& 和 *) …… (181)
8.2.3 指针作为函数的参数 …… (182)
8.2.4 多级指针与指针数组 …… (185)
8.3 指针运算 …… (186)
8.4 指针与数组 …… (188)
8.4.1 指针与一维数组 …… (188)
8.4.2 指针与二维数组 …… (190)
8.4.3 指针与字符数组 …… (192)
8.4.4 指针与函数 …… (195)
8.5 动态存储分配 …… (199)
习题 …… (204)
第 9 章 编译预处理 …… (209)
9.1 宏定义 …… (209)
9.1.1 不带参数的宏定义 …… (209)
9.1.2 带参数的宏定义 …… (211)
9.2 文件包含 …… (214)

9.3 条件编译 …… (216)
习题 …… (218)
第 10 章 结构体和共用体 …… (219)
10.1 结构体类型定义和结构体变量说明 …… (219)
10.1.1 结构体的定义 …… (219)
10.1.2 结构体类型变量的说明 …… (220)
10.2 结构体变量的引用和初始化 …… (222)
10.2.1 结构体类型变量的引用 …… (222)
10.2.2 结构体变量的初始化 …… (223)
10.3 结构体指针变量 …… (226)
10.3.1 指向结构体变量的指针 …… (226)
10.3.2 指向结构体数组的指针 …… (227)
10.3.3 结构体指针变量做函数参数 …… (229)
10.4 常用的内存管理函数 …… (230)
10.5 链表 …… (231)
10.5.1 简单链表的建立 …… (232)
10.5.2 链表的查找 …… (233)
10.5.3 链表的删除 …… (233)
10.5.4 链表的插入 …… (234)
10.5.5 链表的输出 …… (235)
10.6 共用体 …… (238)
10.6.1 共用体的定义 …… (238)
10.6.2 共用体变量的说明 …… (238)
10.6.3 共用体变量的赋值和使用 …… (239)
10.7 枚举类型数据 …… (240)
10.8 用 typedef 定义类型 …… (242)
10.9 综合实例 …… (243)
习题 …… (245)
第 11 章 位运算 …… (249)
11.1 位运算概述 …… (249)
11.1.1 计算机内数据的表示方法 …… (249)
11.1.2 位运算及其运算符 …… (250)
11.2 位运算 …… (250)
11.2.1 按位与 …… (250)
11.2.2 按位或 …… (251)
11.2.3 按位异或 …… (251)
11.2.4 按位取反 …… (252)
11.2.5 按位左移 …… (253)
11.2.6 按位右移 …… (253)

11.2.7 位复合赋值运算符 …… (255)
11.2.8 位运算符的优先级 …… (255)
11.2.9 不同长度的数据进行位运算 …… (255)
11.3 位段 …… (255)
11.3.1 位段的定义 …… (255)
11.3.2 位段的引用 …… (256)
11.4 综合案例分析 …… (258)
习题 …… (260)
第 12 章 文件 …… (262)
12.1 文件概述 …… (262)
12.1.1 文件的基本概念 …… (262)
12.1.2 文件系统 …… (262)
12.1.3 文件的编码方式 …… (263)
12.1.4 文件指针 …… (263)
12.2 文件的打开与关闭 …… (264)
12.2.1 文件的打开(fopen 函数) …… (264)
12.2.2 文件的使用方式 …… (265)
12.2.3 文件的关闭(fclose 函数) …… (266)
12.3 文件的顺序读取 …… (266)
12.3.1 字符的读写函数(fgetc 和 fputc) …… (267)
12.3.2 字符串的读写函数(fgets 和 fputs) …… (268)
12.3.3 格式化的读写函数(fscanf 和 fprintf) …… (269)
12.3.4 数据块的读写函数(fread 和 fwrite) …… (269)
12.4 文件的定位与随机读写 …… (270)
12.4.1 文件定位函数 …… (271)
12.4.2 文件的随机读写操作 …… (271)
12.5 文件的出错检测 …… (272)
习题 …… (273)
附录 A 基本控制字符/字符与 ASCII 值对照表 …… (275)
附录 B C 语言操作符的优先级 …… (276)
参考文献 …… (277)

第1章 C语言程序设计基础

本章介绍了计算机程序设计语言的功能、算法的概念及其描述、C语言的发展历史、C语言的特点、C程序的结构和C程序的上机步骤。学习本章，要求重点掌握算法的描述、C程序的结构和上机运行C程序的方法。学完本章之后，读者将对程序设计以及C语言有一个初步的完整印象。

1.1 程序与程序设计语言

程序是一个非常普遍的概念：为解决某些问题，用计算机可以识别的代码编排的、按照一定的顺序安排的工作步骤。计算机严格按照这些步骤去做，包括计算机对数据的处理。程序的执行过程实际上是对程序所表达的数据进行处理的过程。一方面，程序设计语言提供了一种表达数据与处理数据的功能；另一方面，编程人员必须按照语言所要求的规范（即语法规则）进行编程。

1.1.1 程序与指令

计算机处理数据的基本单元是计算机指令。单独的一条指令本身只能完成计算机的一个最基本的功能，计算机所能实现的指令的集合称为计算机的指令系统。虽然一条计算机指令只能实现一个简单的功能，而且指令系统的指令数目也是有限的，但是一系列有序的指令组合却能完成很复杂的功能。一系列计算机指令的有序组合就构成了程序。

一般情况下，程序的执行是按照指令的排列顺序一条一条地执行的，但是有的程序往往需要通过判断不同的情况，执行不同的指令分支，还有些指令需要被反复执行。

程序在计算机中是以0、1组成的指令码（即机器语言）来表示的，即程序是0、1组成的序列，这个序列能被计算机所识别。一般情况下，程序和数据均存储在存储器中（这种结构称为冯·诺依曼结构，而程序和数据分开存储的结构称为哈佛结构）。当程序要运行的时候，当前准备运行的指令从内存中被调入CPU，由CPU处理该指令。

1.1.2 程序设计语言

语言是人们交换思想的工具，我们日常生活中使用的汉语、英语等称为自然语言。计算机诞生以后，人们要指挥计算机工作就产生了计算机语言。用于程序设计的计算机语言基本上可分为三种：机器语言、汇编语言和高级语言。

1. 机器语言

计算机诞生的初期，人们使用的计算机语言仅由计算机能够识别的0和1代码组成，被称为机器语言。下面是某CPU指令系统中的两条指令：

10000000（进行一次加法运算）

10010000（进行一次减法运算）

用机器语言编程序，就是从所使用的CPU的指令系统中挑选合适的指令，组成一个指

令序列。这种程序虽然可以被机器直接理解和执行，却由于它们不直观、难记、难认、难理解、不易查错，只能被少数专业人员掌握，并且编写程序的效率很低，质量难以保证，这使计算机的推广使用受到了极大的限制。

2. 汇编语言

为减轻人们在编程中的劳动强度，20 世纪 50 年代中期人们开始用一些英文助记符号来代替 0、1 代码编程，于是便产生了符号语言（或称汇编语言）。如前面的两条机器指令可以写为

```
ADD A,B
SUB A,B
```

用汇编语言编程，程序的编写效率及质量都有所提高。但是，汇编语言指令是机器不能直接识别和执行的，而要先翻译成机器语言，程序才能被机器识别和执行。将汇编语言程序转换成二进制代码表示的机器语言的程序（称为汇编程序），经汇编程序“汇编（翻译）”得到的机器语言程序称为目标程序，原来的汇编语言程序称为源程序。由于汇编语言指令与机器语言指令基本上具有一一对应的关系，所以汇编语言源程序的代换可以由汇编系统以查表的方式进行。用汇编语言编写的程序效率高，占用存储空间小，运行速度快，而且用汇编语言能编写出非常优良的程序。

汇编语言和机器语言都不能脱离具体机器即硬件，均是面向机器的语言。不同类型的计算机所用的汇编语言和机器语言是不同的，缺乏通用性，因此，汇编语言被称为低级语言。用面向机器的语言编程，可以编出效率极高的程序，但是程序员用它们编程时，不仅要考虑解题思路，还要熟悉机器的内部结构，并且要“手工”地进行存储器分配，因而其劳动强度仍然很大，给计算机的普及推广造成了很大的障碍。

3. 高级语言

1954 年出现的 FORTRAN 语言以及随后相继出现的其他高级语言，开始使用接近人类自然语言的但又消除了自然语言中的二义性的语言来描述程序。高级语言不受具体机器的限制，使用了许多数学公式和数学计算上的习惯用语，非常擅长于科学计算。用高级语言编写的程序通用性强，直观、易懂、易学，可读性好。到目前为止，世界上有数百种高级语言，常用的有几十种，如 FORTRAN、PASCAL、C、LISP、COBOL 等。这些高级语言使人们开始摆脱进行程序设计必须先熟悉机器的桎梏，把精力集中于解题思路和方法上，使计算机的使用得到了迅速普及。

1.1.3 高级语言程序的开发过程

高级语言程序的开发过程主要包括如下五大步骤。

1. 分析和建立模型

一般来说，一个具体的问题会涉及许多方面，这是问题的复杂性所在。为了便于求解，往往要忽略一些次要方面。这种通过忽略次要方面从而找出解题规律的过程，称为建立模型。

2. 表现模型

表现模型就是用一种符号语言系统来描述模型。一般来说，模型的表现会随着人们对问题抽象程度的加深和细化，不断由领域特色向计算机可解释、可执行的方向靠近（中间也可能采用一些其他的符号系统，如流程图等），直到最后用一种计算机程序设计语言将其描

述出来。

3. 源程序的编辑

源程序的编辑就是在某种字处理环境下，用具体的程序设计语言编写源程序的过程。这不仅要掌握一种计算机程序设计语言，还要应用一种专用程序编辑器或通用的文字编辑器。

4. 程序的编译(或解释)与链接

写出一个高级语言程序后，并不是可以立即拿来执行的。要让机器执行，还要将它翻译成由机器可以直接辨认并可以执行的机器语言程序。为区别它们，把用高级语言编写的程序(文件)称为源程序(文件)，把机器可以直接辨认并执行的程序(文件)称为可执行程序(文件)。这一过程一般分为两步。

第 1 步，在程序编辑过程中输入到源文件中的是一些字符码，但是机器可以直接处理的是 0、1 信息。为此，首先要将源程序文件翻译成 0、1 码表示的信息，并用相应的文件保存。这种保存 0、1 码信息的文件称为目标程序文件。由源文件翻译成目标文件的过程称为编译。在编译过程中，还要对源程序中的语法和逻辑结构进行检查。编译任务是由称为编译器(compiler)的软件完成的。目标程序文件还不能被执行，它们只是一些不连续的目标程序模块。

第 2 步，将目标程序模块以及程序所需的系统中固有的目标程序模块(如执行输入、输出操作的模块)链接成一个完整的程序。经正确链接所生成的文件才是可执行文件。完成链接过程的软件称为链接器(linker)。

图 1-1 为编译和链接过程示意图。

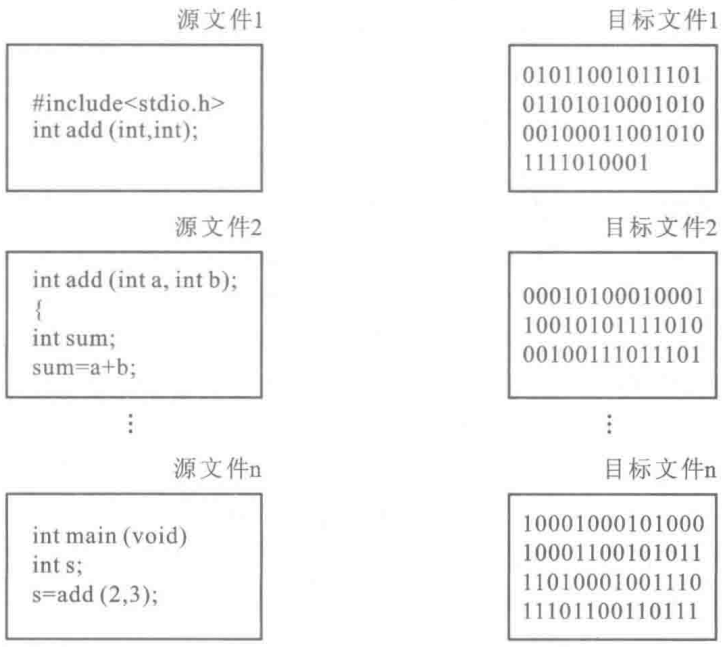

图 1-1　编译和链接过程示意图

5. 程序的测试与调试

程序文件经编译、链接之后，生成可执行文件，这就可以让计算机执行了。但是，并不是一定就可以得到预期的结果，因为程序仍然可能会存在某些错误。因此，每一个人编写出一

个程序后，在正式交付使用前，总要试运行该程序，也就是对程序进行测试。

测试是以程序通过编译后没有语法和链接上的错误为前提的，目的是找出程序中可能存在的错误并加以改正。因此，应该测试程序在不同情况下运行的情况，输入不同的数据可以检测出程序在不同情况下运行的情况。测试的数据应是以“程序是会有错误的”为前提精心设计出来的，而不是随心所欲地乱凑而成的。它们不仅应含有被测程序的输入数据，而且还应包括程序执行它们后所得预期的结果。每次测试都要把实际的结果与预期的结果相比较，以观察程序是否出错。

1.2 C语言概述

1.2.1 C语言的产生与发展

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)，并且他用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。

为了推广UNIX操作系统，1977年Dennis M. Ritchie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》。1978年美国电话电报公司(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，通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准C语言，后来由美国国家标准化协会(American National Standards Institute)在此基础上制定了一个C语言标准，于1983年发表，通常称之为ANSIC。

《K&R》第一版在很多语言细节上不够精确，甚至没有很好表达它所要描述的语言，把后续扩展扔到了一边。C语言在早期项目中的使用受商业和政府合同支配，这意味着一个认可的正式标准是必需的。因此，ANSI于1983年夏天，在CBEMA的领导下建立了X3J11委员会，目的是产生一个C语言标准。X3J11委员会在1989年末提出了“ANSI89”，后来这个标准被ISO接受，即ISO/IEC 9899—1990。

1990年，国际标准化组织ISO(International Organization for Standards)接受了ANSIC89为ISO C的标准(ISO 9899—1990)。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)”。1999年，ISO又对C语言标准进行修订，在基本保留原来C语言特征的基础上，针对应该的需要，增加了一些功能，尤其是对C++中的一些功能，命名为ISO/IEC 9899—1999。2001年和2004年先后进行了两次技术修正。

目前流行的C语言编译系统大多是以ANSI C为基础进行开发的，但不同版本的C语言编译系统所实现的语言功能和语法规则又略有差别。2011年12月，ISO正式公布C语言新的国际标准草案：ISO/IEC 9899—2011。

新的标准提高了对C++的兼容性，并将新的特性增加到C语言中。新功能包括支持多线程，基于ISO/IEC TR 19769—2004规范下支持Unicode，提供更多用于查询浮点数类

型特性的宏定义和静态声明功能。

1.2.2 C语言的特点

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语言可以像汇编语言一样对位、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。

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。另外，C语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

4. C语言是结构式语言

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，这些函数可方便地调用，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。

5. 语法限制不太严格，程序设计自由度大

C语言是强类型语言，但它的语法比较灵活，允许程序编写者有较大的自由度。

6. 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、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，可用来编写系统软件。

7. 生成目标代码质量高，程序执行效率高

一般C语言程序只比汇编程序生成的目标代码效率低10%～20%。

8. 适用范围大，可移植性好

C语言有一个突出的优点就是适合于多种操作系统，如DOS、UNIX、Windows XP、Windows NT，也适用于多种机型。C语言具有强大的绘图能力，可移植性好，并具备很强的数据处理能力，因此适于编写系统软件，三维、二维图形和动画。

1.2.3 C语言的应用

C语言的特长不在科学计算和管理领域。对操作系统和系统应用程序以及需要对硬件进行操作的场合，使用C语言会明显地优越于使用其他高级语言。C语言是当前比较流行的一种编程语言，常被用于系统软件和应用软件的开发之中。

下面简单介绍C语言应用较多的几个方面。

1. 数据库管理系统及应用程序方面

C语言具有汇编语言的特点，比较适合编写系统软件，因而常被广泛用于开发数据库管理系统和应用软件。在很长一段时间里，大多数关系数据库管理系统软件都是用C语言开发的，如dBASE、FoxBASE、ORACLE；大多数数据库系统软件也是用C语言开发的。目前，随着面向对象语言的发展，C语言主要用于实现数据库与前台之间的连接。

2. 图形图像系统和应用程序方面

C语言在图形图像的开发中有着广泛的应用。很多图形图像系统软件包都是采用C语言编写的。例如，被广泛使用的通用图形软件系统AutoCAD就是用C语言开发的，并直接支持C语言程序。C语言编译系统本身带有许多具有绘图功能的函数，利用这些函数开发图形应用软件十分方便，如许多人直接使用C语言编译系统提供的绘图环境实现不同领域的专业绘图设计。

3. 编写与设备的接口程序方面

C语言在创建友好的交互式图形界面上有着广泛应用。使用C语言可以方便地实现下拉式菜单、弹出式菜单和多窗口技术等功能，并且在编写设备接口程序方面也有着广泛的应用。通常人们喜欢用汇编语言编写设备接口程序，这样效率较高。由于C语言既具有高级语言的特性又具有汇编语言的部分功能，因此，使用C语言和汇编语言混合编写接口程序也很方便。

4. 数据结构方面

C语言本身提供了十分丰富的数据类型，包括基本数据类型和构造数据类型。使用C语言提供的数据类型可以很方便地解决复杂的数据结构问题。例如，可以方便地编写关于链表结构、队列结构、栈结构以及树结构的程序，而且在许多方面也有成熟的程序供选择使用。

5. 排序和检索方面

在大量的数据处理问题中，排序和检索是重要的处理方法。使用C语言来编写排序和检索程序既方便又简洁，因为C语言支持递归算法，使用递归函数编写排序程序，程序显得清晰明了。所以，使用C语言进行繁杂的数据处理有时会得心应手。

以上列举了C语言在五个方面的应用，而C语言的应用远不止这些。

1.3 C语言开发程序

1.3.1 用C语言开发程序的过程

C语言是一种高级计算机语言，是人们借助计算机解决问题的一种工具。使用C语言编写程序的最终目的就是解决实际问题。既然要解决实际问题，显然只有工具是不够的，我们需要有效地、合理地使用工具才能更好地解决实际问题。

使用C语言解决实际问题时，通常分为以下几个阶段：

(1) 分析问题；

(2) 设计算法；

(3) 编写代码；

(4) 编译与调试；