



普通高等教育“十三五”规划教材·计算机系列

C语言

程序设计（第二版）

叶斌 陈世强 主编



科学出版社

普通高等教育“十三五”规划教材·计算机系列

C 语言程序设计

(第二版)

叶 斌 陈世强 主 编

贺 刚 陈自根 副主编

科学出版社

北 京

内 容 简 介

本书对 C 语言的基本构成、语法规则、使用特性及程序设计的基本方法与技术做了系统而详细的介绍。全书注重算法设计与程序设计的有机结合,强调模块化程序设计思想及其实现方法,强化工程应用训练。

本书共 10 章,内容分别为 C 语言及程序设计概述,基本数据类型和表达式,程序的控制结构,函数,数组,指针,结构体、共用体和枚举类型,文件,位运算及综合应用案例。为了使读者更好地掌握各章节内容,每章还配有大量精选的习题。

本书可作为高等院校各专业计算机程序设计课程的教学用书,也可作为培训教材或一般工程技术人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计/叶斌,陈世强主编. —2 版. —北京:科学出版社,2018.7
普通高等教育“十三五”规划教材·计算机系列

ISBN 978-7-03-057388-9

I. ①C… II. ①叶… ②陈… 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(2018)第 097767 号

责任编辑:戴薇 王惠 / 责任校对:王万红

责任印制:吕春珉 / 封面设计:东方人华平面设计部

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>

三河市良远印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2012 年 6 月第 一 版 开本:787×1092 1/16

2018 年 7 月第 二 版 印张:22 1/4

2018 年 7 月第一次印刷 字数:524 000

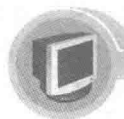
定价:56.00 元

(如有印装质量问题,我社负责调换〈良远〉)

销售部电话 010-62136230 编辑部电话 010-62135397-2052

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303



C 语言是一种结构化程序设计语言。它既有高级语言的特点，又具有汇编语言的许多功能；既适于编写应用软件，又能用于开发系统软件，是目前功能强大、应用广泛、影响力强的程序设计语言之一。因此，C 语言是计算机从业人员和计算机爱好者都希望掌握的程序设计语言。

本书在全面而系统地介绍 C 语言的基本概念、语法功能、使用特性和程序设计方法及技能的基础上，通过具体案例分析，由浅入深，循序渐进，力求使读者理解和掌握数据结构、算法的概念及其应用；着重培养读者良好的程序设计思想和编程能力，为其进一步学习、从事软件开发和工程应用打下坚实的基础。全书由 4 个部分共 10 章组成，具体内容组织如下。

第 1 部分为程序设计基础，由第 1 章和第 2 章组成，主要介绍 C 语言的发展和特点，C 语言程序的基本结构、编码风格，C 语言程序的开发过程与环境，基本数据类型、运算符与表达式等。这部分内容是 C 语言程序设计的入门基础知识。

第 2 部分为程序设计的基本结构，只含第 3 章，主要介绍算法及其描述方法、简单算法的设计、C 语言的语句分类、程序的流程控制结构等。这部分内容主要是让读者了解算法的作用及构造；掌握结构化程序设计的三种基本结构：顺序结构、选择结构和循环结构，以及实现这些结构的语句。

第 3 部分为模块化程序设计，由第 4~9 章组成，主要介绍 C 语言的模块化程序设计思想和实现方法、指针类型及构造类型、文件操作等。这部分内容主要是让读者全面了解和掌握 C 语言程序设计的方法和手段，培养读者的程序设计思想和编程能力，使其学会用计算思维方式去思考问题和解决问题。

第 4 部分为一个综合应用案例，只含第 10 章，主要介绍一个与手机电话本功能类似的电话本软件项目的开发过程。这部分内容旨在让读者进一步巩固、理解和综合运用所学知识，从工程应用的角度了解利用 C 语言进行软件开发的方法与过程，从而达到综合训练、提高能力的目的。

本书具有以下特色。

1. 以算法为核心，以程序设计思想为主线。

本书在编写理念上突出以算法为核心，以程序设计思想为主线。书中的案例不仅给出了例程，而且对算法设计及编程方法与技巧进行了较详细的解析。

2. 案例典型，富于启发性。

书中精选的案例类型丰富，具有代表性。这些案例都提供了源程序并且都已在 Visual C++ 6.0 环境下调试通过，运行结果正确。例程极富启发性，能激发读者积极思考，寻求解决问题的新方法。

3. 代码规范, 风格一致。

编者在编写本书的过程中十分重视对读者良好编程风格的培养。代码书写规范、统一; 程序版式追求清晰、美观; 注释的运用合理、准确, 容易理解。编码风格力求与 Windows 应用程序风格保持一致。例如, 标识符的命名规则在参考匈牙利命名法的基础上, 统一采用驼峰式命名法。

4. 以工程应用为目的, 注重编程实训。

编者在编写本书的过程中十分重视以工程应用为目的, 以培养工程应用型人才为目标。例如, 本书第 10 章的综合应用案例正是围绕此目标设置的。本书以实例导入知识点, 采用案例驱动的方式, 强调理论与实践相结合, 注重编程实训及培养读者的综合应用能力和软件开发能力。

5. 习题丰富典型, 且配有实验指导与习题解答。

为了帮助读者学习和巩固 C 语言理论知识, 书中每章都附有习题, 通过这些练习和上机编程训练, 可以使读者加深理解程序设计的基本思想、掌握编程的基本方法和技巧、提高程序设计能力。与本书配套的《C 语言程序设计实验指导与习题解答(第二版)》也将同步出版, 其中给出了习题的参考答案, 安排了上机实验内容, 可有效提升读者学习效果。

6. 附录内容丰富, 实用性强。

本书安排了 5 个附录, 内容丰富, 实用性强, 便于教学和学习时查找相关内容。

本书由叶斌、陈世强担任主编, 由贺刚、陈自根担任副主编。所有编者都是多年在教学一线从事“C 语言程序设计”课程教学及计算机软件教学的老师, 具有丰富的教学经验和较强的软件开发能力。本书第 1~4 章、第 9 章由叶斌编写, 第 5 章、第 8 章、第 10 章由贺刚编写, 第 6 章、第 7 章由陈自根编写, 附录由贺刚、陈自根编写。全书由叶斌、陈世强负责统稿。

本书的出版得到了学校各级领导的热切关心和大力支持。在本书的编写过程中, 沈济南、胡俊鹏老师参与了内容校对工作并提出了许多有益的建议; 同时, 本书的编写中参考了许多同行的著作, 在此一并表示衷心的感谢。

尽管编者以高度的责任心和百倍的努力投入写作, 但由于学识水平所限, 书中难免存在疏漏和不妥之处, 恳请广大读者批评指正。

编 者

2018 年 5 月



第 1 章 C 语言及程序设计概述	1
1.1 程序与程序设计语言	1
1.1.1 程序的概念	1
1.1.2 程序设计语言的发展	1
1.1.3 C 语言的发展及特点	3
1.1.4 C 语言与 C++、Java、C# 的比较	5
1.2 程序设计方法	6
1.2.1 程序设计的基本过程	6
1.2.2 结构化程序设计方法	6
1.2.3 面向对象程序设计方法	7
1.3 C 语言程序的基本结构	8
1.3.1 一个简单的 C 语言程序	8
1.3.2 C 程序的结构特点	9
1.3.3 代码编写风格	10
1.4 C 语言的基本语法单位	10
1.4.1 C 语言的字符集	10
1.4.2 关键字	11
1.4.3 标识符	11
1.4.4 分隔符	12
1.4.5 注释	12
1.5 C 语言程序的开发过程与环境	13
1.5.1 由源程序生成可执行程序的过程	13
1.5.2 Visual C++ 6.0 集成开发环境	14
1.5.3 Code::Blocks 集成开发环境	17
习题 1	19
第 2 章 基本数据类型和表达式	22
2.1 基本数据类型	22
2.1.1 整型数据	22
2.1.2 实型数据	24
2.1.3 字符型数据	24
2.2 常量与变量	26

2.2.1 常量	27
2.2.2 变量	29
2.3 运算符与表达式	32
2.3.1 算术运算符与算术表达式	32
2.3.2 自增与自减运算	34
2.3.3 关系运算符与关系表达式	35
2.3.4 逻辑运算符与逻辑表达式	36
2.3.5 赋值运算符与赋值表达式	38
2.3.6 条件运算符与求字节运算符	39
2.3.7 逗号运算符与逗号表达式	40
2.4 数据类型转换	40
2.4.1 自动类型转换	40
2.4.2 赋值运算时的类型转换	41
2.4.3 强制类型转换	42
习题 2	42
第 3 章 程序的控制结构	46
3.1 算法与语句	46
3.1.1 算法及其特征	46
3.1.2 算法与程序结构	47
3.1.3 算法的描述	47
3.1.4 C 语言的语句分类	52
3.2 基本输入/输出函数	54
3.2.1 字符输入/输出函数	54
3.2.2 格式化输入/输出函数	55
3.3 顺序结构	60
3.4 选择结构	61
3.4.1 if 语句	61
3.4.2 if 语句的嵌套	65
3.4.3 switch...case 语句	68
3.5 循环结构	70
3.5.1 while 语句	70
3.5.2 do...while 语句	72
3.5.3 for 语句	74
3.5.4 循环嵌套	77
3.6 其他控制语句	79
3.6.1 break 语句	80

3.6.2	continue 语句	80
3.6.3	goto 语句	81
3.7	程序设计举例	82
	习题 3	86
第 4 章	函数	93
4.1	结构化程序设计与 C 程序结构	93
4.1.1	结构化程序设计的特征与风格	93
4.1.2	模块与函数	94
4.2	标准库函数与函数的定义	95
4.2.1	标准库函数	95
4.2.2	函数的定义	96
4.3	函数的一般调用	98
4.3.1	函数的声明	98
4.3.2	函数的调用	99
4.3.3	参数传递	100
4.3.4	函数的返回值	102
4.4	函数的嵌套调用与递归调用	104
4.4.1	函数的嵌套调用	104
4.4.2	函数的递归调用	105
4.5	变量的作用域	107
4.5.1	局部变量	107
4.5.2	全局变量	109
4.6	变量的存储类别	111
4.6.1	变量的存储方式	111
4.6.2	自动变量	112
4.6.3	静态变量	112
4.6.4	寄存器变量	113
4.6.5	外部变量	114
4.7	内部函数与外部函数	116
4.7.1	内部函数	116
4.7.2	外部函数	116
4.8	编译预处理	117
4.8.1	编译预处理简介	117
4.8.2	宏定义	118
4.8.3	文件包含	121
4.8.4	条件编译	122

4.9 程序设计举例	125
习题 4	129
第 5 章 数组	135
5.1 一维数组	135
5.1.1 一维数组的定义	135
5.1.2 一维数组的逻辑结构和存储结构	137
5.1.3 一维数组元素的引用	137
5.1.4 一维数组的初始化	140
5.1.5 一维数组的应用举例	141
5.2 二维数组	144
5.2.1 二维数组的定义	145
5.2.2 二维数组的逻辑结构和存储结构	145
5.2.3 二维数组元素的引用	146
5.2.4 二维数组的初始化	148
5.2.5 二维数组的应用举例	149
5.3 字符数组和字符串	152
5.3.1 字符数组的定义和初始化	152
5.3.2 字符数组的输入/输出	155
5.3.3 字符串的概念和存储表示	157
5.3.4 字符串处理函数	159
5.4 数组作为函数的参数	164
5.4.1 数组元素作为函数的参数	164
5.4.2 数组名作为函数的参数	165
5.5 程序设计举例	167
习题 5	171
第 6 章 指针	176
6.1 指针概述	176
6.1.1 变量的地址	176
6.1.2 指针和指针变量	177
6.1.3 指针变量的定义	177
6.1.4 指针变量的初始化	178
6.1.5 指针变量的引用	178
6.2 指针运算	180
6.2.1 指针的赋值运算	180
6.2.2 指针的算术运算	181

6.2.3 指针的关系运算	182
6.2.4 指针的下标运算	182
6.3 指针与函数	183
6.3.1 指针作为函数的参数	183
6.3.2 返回指针的函数	184
6.3.3 指向函数的指针	185
6.4 指针与数组	187
6.4.1 指向数组元素的指针	188
6.4.2 指向一维数组的指针	190
6.4.3 指针数组	191
6.4.4 多级指针	192
6.5 指针与字符串	194
6.5.1 字符型指针与字符串	194
6.5.2 字符串处理函数的实现	195
6.5.3 字符串数组	197
6.5.4 带参数的 main()函数	199
6.6 程序设计举例	200
习题 6	206
第 7 章 结构体、共用体和枚举类型	212
7.1 结构体	212
7.1.1 结构体类型的声明	212
7.1.2 结构体变量的定义	214
7.1.3 结构体变量的引用	216
7.1.4 结构体变量的初始化	216
7.1.5 结构体变量的有关操作	217
7.1.6 结构体数组	219
7.1.7 结构体指针变量	222
7.1.8 结构体与函数	224
7.2 共用体	229
7.2.1 共用体类型声明及共用体类型变量的定义	229
7.2.2 共用体变量的引用	230
7.2.3 共用体变量的初始化	231
7.3 枚举类型	233
7.3.1 枚举类型的声明	233
7.3.2 枚举类型变量的定义	234
7.4 用 typedef 定义类型	236

7.4.1	typedef 的意义	236
7.4.2	typedef 的用法	236
7.5	链表	237
7.5.1	单链表的构造	237
7.5.2	单链表的操作	239
7.6	程序设计举例	247
习题 7		253
第 8 章	文件	262
8.1	文件概述	262
8.1.1	文件的基本概念	262
8.1.2	文件的分类	262
8.1.3	文件缓冲区	263
8.2	文件类型指针	264
8.3	文件的打开与关闭	265
8.3.1	文件的打开	265
8.3.2	文件的关闭	267
8.4	文件的读/写操作	267
8.4.1	字符读/写函数	267
8.4.2	字符串读/写函数	271
8.4.3	数据块读/写函数	272
8.4.4	格式化读/写函数	275
8.5	文件的随机读/写操作	276
8.5.1	重返文件头函数	276
8.5.2	指针位置移动函数	277
8.5.3	检测指针当前位置函数	279
8.5.4	文件操作出错检测函数	279
8.5.5	文件处理范例	280
习题 8		285
第 9 章	位运算	288
9.1	位运算符与位运算	288
9.1.1	按位取反运算符	288
9.1.2	左移运算符	289
9.1.3	右移运算符	290
9.1.4	按位与运算符	291
9.1.5	按位或运算符	292

9.1.6 按位异或运算符	293
9.2 位段	294
9.2.1 位段结构体的说明	294
9.2.2 位段的引用	295
9.3 程序设计举例	296
习题 9	297
第 10 章 综合应用案例	300
10.1 系统设计要求	300
10.2 系统设计及函数实现	301
10.2.1 系统设计	301
10.2.2 数据结构	302
10.2.3 函数设计	303
10.3 参考程序	312
10.3.1 源代码清单	312
10.3.2 电话本软件开发过程简介	327
习题 10	327
附录 A ASCII 码表	329
附录 B C 语言的关键字及说明	330
附录 C 运算符的优先级和结合性	331
附录 D 常用的 C 语言库函数	332
附录 E 用户自定义标识符的命名规则	340
参考文献	342

第1章 C语言及程序设计概述

C语言是一种结构化的计算机程序设计语言，应用范围广泛。它既具有高级语言的特点，又具有汇编语言的某些功能。它可以作为系统设计语言，用来编写系统应用程序；也可以作为应用程序设计语言，用来编写不依赖计算机硬件的应用程序。本章从程序的概念及程序设计语言的发展入手，介绍C语言的基本特点、C语言程序的基本结构、C语言基本语法单位、C语言程序的开发过程及集成开发环境（integrated developing environment, IDE）Visual C++ 6.0 和 Code::Blocks。

本章内容是学好C语言程序设计的基础，将用一个简单范例让读者实际感受C语言程序的结构与特点，基本理解C语言程序设计必须遵循的规范，掌握编写和调试一个C语言程序的方法，这是每一个C程序员所必须具备的基本功。

1.1 程序与程序设计语言

1.1.1 程序的概念

计算机程序是指可以被计算机或其他信息处理装置连续执行的一条条指令的集合。也就是说，程序（program）是能够完成特定任务的指令序列。

我们知道，指令是二进制码，用它编制程序既不便记忆，又难以掌握。于是，计算机科学家就研制出了多种便于人们理解和使用的计算机语言，如汇编、C/C++、Java、C#等语言。这些计算机语言通常被称为程序语言。

用某种程序语言编制出来的源程序文件一般要经过编译和连接后得到可执行的程序文件（扩展名一般是.exe），如图1-1所示。

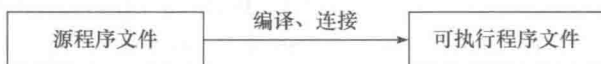


图 1-1 计算机源程序和可执行程序的关系

❗ 注意：

本书所说的编程是指为解决某个问题而使用某种程序设计语言编写源程序、调试、编译、连接得到可执行程序的全部过程。

1.1.2 程序设计语言的发展

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历程。

1. 机器语言

机器语言 (machine language) 或称为二进制代码语言, 是一串串由“0”和“1”组成的指令, 计算机可以直接识别, 不需要进行任何翻译。对于每台机器的指令, 其格式和代码所代表的含义都是硬性规定的, 故称之为机器语言。不同型号的计算机的机器语言一般是不同的。直接使用机器语言编程是非常辛苦的, 效率低且容易出错, 同时要求编程人员非常熟悉计算机硬件。机器语言是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦, 人们进行了一种有益的改进: 用一些简洁的英文字母、符号串 (称为指令助记符) 来替代一个特定指令的二进制串。例如, 用“ADD”代表加法, 用“MOV”代表数据传递等。这样, 人们就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 (assembly language), 即第二代计算机语言。

但是, 计算机是不认识这些助记符号的, 因此需要一个专门的程序, 负责将这些符号翻译成二进制的机器语言, 这种翻译程序称为汇编程序。

汇编语言同样十分依赖于机器硬件, 移植性不好, 但效率十分高, 针对计算机特定硬件而编制的汇编语言程序能准确发挥计算机硬件的功能和特长, 程序精练且质量高, 所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中, 人们意识到, 应该设计一种这样的语言: 接近数学语言或人的自然语言, 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, 1954年, 第一个完全脱离机器硬件的高级语言 (high-level programming language) ——FORTRAN 问世了。60多年来, 共有几百种高级语言出现, 影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、NOBOL、PL/1、Pascal、C、Prolog、Ada、C++、Visual C++、Visual Basic、Delphi、Java、C#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, 从面向过程语言 (procedure oriented language) 到面向对象语言 (object oriented language) 的历程。FORTRAN、ALGOL、COBOL、BASIC、LISP、NOBOL、PL/1、Pascal、C、Prolog、Ada 等是面向过程语言, 而 C++、Visual C++、Visual Basic、Delphi、Java、C#等是面向对象语言。

高级语言的下一个发展目标是面向应用, 即只需要告诉程序你要干什么, 程序就能自动生成算法, 自动进行处理, 这就是非过程化的程序语言。

为了让读者了解机器语言、汇编语言和高级语言程序设计的复杂度, 下面以求 1+1 的值为例进行说明, 如图 1-2 所示。

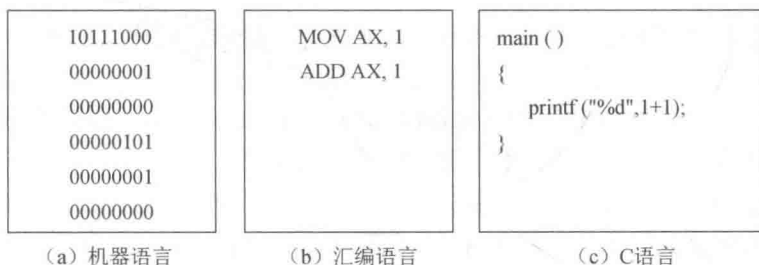


图 1-2 求 1+1 问题的机器语言、汇编语言、高级语言 (C 语言) 程序

1.1.3 C 语言的发展及特点

1. C 语言的发展概况

C 语言是 1972 年由美国的 D.M.Ritchie (1983 年获得图灵奖, 1999 年获得美国国家技术奖) 设计发明的, 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。同年, 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, 后来美国国家标准协会 (American National Standards Institute, ANSI) 在此基础上制定了一个 C 语言标准, 于 1983 年发表, 通常称为 ANSI C 83。

1987 年, 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, 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, ANSI 为 C 语言制定了一套新的 ANSI 标准——ANSI C 87, 目前流行的 C 编译系统都是以此为基础的。1990 年, 国际标准化组织 (International Standard Organization, ISO) 将 ANSI C 87 采纳为 ISO C 语言的标准 (ISO/IEC 9899:1990 Programming languages—C)。

1999 年, ANSI 和 ISO 发布了新版本的 C 语言标准和技术勘误文档, 该标准称为 ANSI C 99。ANSI C 99 标准之后, 新的 C 语言标准是 ISO 和国际电工委员会 (International Electrotechnical Commission, IEC) 在 2011 年 12 月 8 日正式发布的 ANSI C 11 标准, 官方正式名为 ISO/IEC 9899:2011。它基本上是目前关于 C 语言最新、最权威的定义。现在, 各种 C 语言编译器都提供了对 ANSI C 87 的完整支持, 对 ANSI C 99 只提供了部分支持; 另外, 还有一部分 C 语言编译器提供了对某些 K&R C 风格的支持。

目前, 常见的 C 语言编译开发工具有 Visual C++、Borland C++、Borland C++ Builder、Watcom C++、GNU DJGPP C++、LCC Win32 C、High C、C-Free、Turbo C 等, 这些工具大多遵循 ANSI C 87 的规范, 只是在某些细节上存在差异。在 Windows 操作系统上以 Visual C++ 6.0 最为常用。

2. C 语言的特点

C 语言是一种结构化语言, 层次清晰, 便于按模块化方式组织程序, 易于调试和维护。C 语言的表现能力和处理能力极强, 它不仅具有丰富的运算符和数据类型, 还可以

直接访问内存的物理地址, 进行位操作。具体来讲, C 语言的特点如下:

(1) 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字 (ANSI C 87)、9 种控制语句, 它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, 编程自由灵活。

(2) 运算符丰富

C 语言的运算符很丰富, 共有 44 个运算符。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运算, 并引入了指针概念, 使程序效率更高。

(4) 结构化的程序设计语言

C 语言是一种结构化语言, 提供了编写结构化程序的基本控制语句, 并以具有独立功能的函数形式作为模块化程序设计的基本单位提供给用户。这些函数可以方便调用, 有利于利用模块化方式进行程序设计、编码、调试和维护。这种结构化方式可使程序层次清晰, 程序的各个部分除了必要的信息交流外彼此独立。

(5) 语法限制不太严格, 程序设计自由度大

一般的高级语言语法检查比较严格, 能够检查出大多数的语法错误, 而 C 语言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, 可以直接对硬件进行操作

C 语言既具有高级语言的特点, 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, 而这三者是计算机基本的工作单元, 可以用来编写系统软件。

(7) 程序生成代码质量高, 程序执行效率高

C 语言程序一般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、Linux、Windows 等, 同时, 其适用于多种机型。

当然, C 语言也有自身的不足。例如, 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, 对数组下标越界不做检查等。从应用的角度来看, 与其他高级语言相比, C 语言较难掌握。

总之, C 语言既具有高级语言的功能, 又具有汇编语言的特点; 既是一个成功的系统设计语言, 又是一个高效的应用程序设计语言; 既能用来编写不依赖计算机硬件的应用程序, 又能用来编写各种系统程序。由于 C 语言的这些突出特点, 它的应用领域非常广泛, 这里列出一些典型的应用领域, 以指导读者今后的学习和选择。

1) C 语言适合用于开发系统软件和大型应用软件, 如操作系统、编译系统、高性能应用服务器软件等。

2) 在软件需要对硬件进行操作的场合,用C语言明显优于其他高级语言。例如,各种硬件设备的驱动程序(如网卡驱动程序、显卡驱动程序、打印机驱动程序等)。

3) 在图形、图像及动画处理方面,C语言具有绝对优势。例如,游戏软件的开发可使用C语言。

4) 适合编写网络通信程序。随着计算机网络的飞速发展,特别是Internet的出现,分布式软件间的通信显得尤其重要,而通信程序的编写首选就是C语言。

5) 嵌入式系统开发中主要使用C语言。在此之前,人们主要使用汇编语言。

6) C语言适用于跨操作系统平台的软件开发。Windows、UNIX、Linux、OS/2等绝大多数操作系统支持C语言,其他高级语言未必能得到支持,所以编写在不同操作系统下运行的软件用C语言是最佳选择。

1.1.4 C语言与C++、Java、C#的比较

尽管C语言与C++、Java、C#不是同一种语言,但是它们之间也有所联系。从广义上讲,C语言可以看作其他三种语言的源语言,这是因为无论是从数据类型还是从控制语句来看,其他三种语言都有来自C语言的迹象。

一般将C++看作对C语言的扩展。因为C语言没有面向对象的语法结构,而当时业界又迫切需要面向对象的编程特性,所以贝尔实验室的开发者就为C语言添加了面向对象的结构。现在C++已经不只是C语言的扩展了,它已经完全可以看作一种新的编程语言。虽然C语言的特性及库函数仍然被C++支持,但是C++已拥有自己独立的类库体系,功能相当强大。

Java是一种完全面向对象的语言,虽然它的底层(运行时库)是用C语言开发的,但其并不依赖于C语言。Java的运行是在运行时库的支持下进行的,所以运行效率相对于可以更接近底层的C/C++会有所降低,但是Java的类库采用了很好的设计理念,非常好用,也非常实用,已经成为业界的一种标准开发语言。它的跨平台特性受到很多开发者的欢迎,用户只需要开发一次就能在所有安装了Java运行时库的系统上运行。

C#是Microsoft公司开发的一种编程语言,语法类似Java,几乎就是Java的翻版。其运行原理和Java类似,也是通过运行时库的支持运行,但是支持的平台很有限。Java被大多数平台支持,而C#目前只被Windows和Linux支持,Windows下的支持是由Microsoft公司自己完成的,而Linux下的支持则由Mono完成。实际上,Mono只是把C#应用转化为Java应用而已,所以本质上,C#仍然只是被Microsoft公司自己的操作系统支持。

C/C++的优点在于与底层比较接近,可以控制的粒度更加精细,是开发系统级应用的最佳选择。C/C++的缺点源于其优点,因为它们能控制的编程元素粒度精细,所以编程比较困难,容易出错。Java和C#都比较高,可以看作高级语言的高级语言,优点是开发容易,但运行效率不如更为接近底层的C/C++。

在具体工程应用中,可以根据实际的项目需要来选择编程语言。对于运行效率要求高的、底层控制要求高的项目用C/C++编程,否则可选择Java或C#编程;对于跨平台