



普通高等教育“十一五”国家级规划教材

游洪跃 许春 谭斌 主编
伍良富 李炳法 主审

C语言程序设计教程

21世纪计算机科学与技术实践型教程

丛书主编 陈明



清华大学出版社



普通高等教育“十一五”国家级规划教材

游洪跃 许春 谭斌 主编
伍良富 李炳法 主审

C语言程序设计教程

21世纪
计算机
科学

与实践
型教程

丛书主编
陈明

清华大学出版社
北京

内 容 简 介

全书共9章,阐述了C语言的主要特点及C++程序开发过程,介绍了基本数据类型、表达式和运算符,结构化程序控制流程,函数的使用以及相关知识,指针和数组的使用方法,用户定制类型及位运算,预处理器的使用方法,文件的基本操作方式,以及一些关于C语言的高级内容。

本书覆盖了二级计算机等级考试大纲《C语言程序设计》的全部考点,每章都有“相关知识”,包含了大部分二级等级考试公共基础知识,通过潜移默化方式让读者接受二级等级考试公共基础知识,每章习题中的选择题和填空题全部改编于全国计算机二级等级考试原题,各章的编程习题一般来源于真实考试题,所有习题稍加修改都可作为考试试题。

通过本书的学习,读者能迅速提高C语言程序设计的能力,经过适当的选择,本书可作为C语言程序设计课程的教材,本书可作为高等院校计算机及相关专业的教材,也可供其他从事软件开发工作的读者参考使用,本书不但适合高等学校学生使用,同时也适合初学程序设计者或有一定编程实践基础、希望突破编程难点的读者作为自学教材。

本书全面系统地介绍了C语言程序设计各要素,取材新颖,内容丰富,可读性强。本书充分考虑了读者对书中部分内容的心理适应性,对于一些容易让读者产生畏惧心理的内容作了适当的处理。本书所有程序都在Visual C++ 6.0、Dev-C++和MinGW Developer Studio开发环境中进行了严格的测试,在作者教学网站上提供了大量的教学支持内容。

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语言程序设计教程/游洪跃,许春,谭斌主编. —北京:清华大学出版社,2011.6
(21世纪计算机科学与技术实践型教程)
ISBN 978-7-302-23046-5

I. ①C… II. ①游… ②许… ③谭… 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(2010)第113419号

责任编辑:汪汉友

责任校对:李建庄

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:19.5

字 数:485千字

版 次:2011年6月第1版

印 次:2011年6月第1次印刷

印 数:1~3000

定 价:31.00元

产品编号:037636-01

《21 世纪计算机科学与技术实践型教程》

编辑委员会

主 任：陈 明

委 员：	毛国君	白中英	叶新铭	刘淑芬	刘书家
	汤 庸	何炎祥	陈永义	罗四维	段友祥
	高维东	郭 禾	姚 琳	崔武子	曹元大
	谢树煜	焦金生	韩江洪		

策划编辑：谢 琛

《21 世纪计算机科学与技术实践型教程》

序

21 世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。`国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细地思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

前 言

作者使用过国内数本 C 语言程序设计的教材,都不十分满意,C 程序教学的普遍结果是,学生学完了 C 语言,但却不会使用目前流行的 C 开发工具编写程序。同时几乎所有教材都存在错误。例如某经典教材的典型例题程序如下:

```
#include <stdio.h>                                /* 包含库函数 printf()所需要的信息 */

void CopyString(char *from, char *to)
{
    for(; *from!='\0'; from++, to++) *to= *from;    /* 复制 from 到 to */
    *to= '\0';                                     /* 加上字符串结束符 */
}

int main(void)                                     /* 主函数 main() */
{
    char *a="I am a teacher.";                    /* 定义字符指针 */
    char *b="You are a student.";                  /* 定义字符指针 */

    printf("a:%s\nb:%s\n", a, b);                 /* 显示字符串 a,b */
    CopyString(a, b);                              /* 复制 a 到 b */
    printf("a:%s\nb:%s\n", a, b);                 /* 显示字符串 a,b */

    return 0;                                       /* 返回值 0, 返回操作系统 */
}
```

上面的程序在当前流行 C/C++ 编程器 Visual C++ 6.0 SP6, Dev-C++ 4.9.9.2 和 MinGW Developer Studio 2.05 都能正常通过编译,但运行到函数调用“CopyString(a, b);”时将出现运行错误,实际上函数 CopyString()本没问题,其错就错在实参上,它试图把一个字符串常量赋值给另一个字符串常量。这在概念上就是错误的。书籍中存在错误是在所难免的,但这种潜在错误对读者的影响还是很难估计。由于读者很难有机会发现这种错误,即或是发现了这类错误,可能读者还以为自己的理解错误了,这样读者会一直延续这种错误的观念,再把这种观念带到实际编程工作中,带来的社会影响就更大。这个问题在某教材中存在着很多年,直到最近的最新版也依然存在。这也从侧面说明了改变这样的错误是要花时间代价的。

再比如类似下面的程序：

```
#include <stdio.h>                                /* 包含库函数 printf() 所需要的信息 */
void main(void)                                    /* 主函数 main() */
{
    printf("Hello, world!\n");                      /* 输出 "Hello, world" */
}
```

在 Visual C++ 6.0 SP6 编译器下可以编译运行,而在语法检查更严格的 Dev-C++ 4.9.9.2 和 MinGW Developer Studio 2.05 下根本通不过,这是由于在新标准中 main() 函数返回值类型不能为 void,这样的程序即使在部分编译器上能够正常编译运行,但也是错误的,这类程序已根本谈不上可移植性了。

本书作者在经过十多年教学及查阅大量参考资料后编写了本书。全书共 9 章,第 1 章重点阐述 C 语言的主要特点及 C++ 程序开发过程;第 2 章着重介绍了基本数据类型、表达式和运算符;第 3 章介绍了结构化程序控制流程,其中重点讨论了 3 种基本控制结构;第 4 章着重探讨了函数的使用以及相关知识;第 5 章讨论了指针和数组的使用方法;第 6 章讨论了用户定制类型及位运算,包括结构、联合、位运算和枚举;第 7 章阐明了预处理器的使用方法;第 8 章介绍了文件的基本操作方式;第 9 章是一些关于 C 语言高级话题的讨论。

对于初学者,考试时往往会感到茫然而不知所措,因此本书习题包括了选择题、填空题和编译题,这些题目选自于考试题,可供学生期末复习,也可供教师出试题时参考,对教师还提供了习题参考答案。

全书各章都提供有“相关知识”,这些经过精心选择的相关知识包含了大部分二级等级考试公共基础知识,根据美国计算机学会(ACM)的课程建议,在计算机专业基础课程应帮助学生树立对学科各个方面的整体认识,进而为计算机科学后续课程的学习打下坚实的基础,本书“相关知识”包含了大部分二级等级考试所有公共基础知识,通过潜移默化方式让学生接受二级等级考试公共基础知识,这些知识都尽量以通俗易懂的方式进行介绍,为读者参考全国计算机二级等级考试提供帮助。

本书在部分章节中还提供了实例研究,主要提供给那些精力充沛的学生深入学习与研究,每个实例研究都有一定的目的与意义,包括了对“相关知识”的进一步的实现的实例研究(例如第 5 章的实例研究“顺序查找”、“冒泡排序”,第 6 章实例研究“顺序栈”,这些实例研究都属于“相关知识”的进一步实现,基础稍好的学生都能通过自学有所收获,当然老师也可选讲或全部进行讲解),还包括了算法设计的实例研究(例如第 4 章的实例研究“汉诺塔问题”,虽然难度一般,但确是算法设计中关于“递归”算法的实例,为将来学习算法设计打下坚实的基础,也为 C++ 或数据结构学习关于“递归”课程设计项目作伏笔),以及包括了“相关知识”的综合应用的实例研究(例如第 8 章的实例研究“人事管理系统”,采用“软件工程”的方法进行分析,实现了一个简单的“数据库管理系统”)。

为了尽快提高读者的实际编程能力,本书各章提供了程序陷阱,包含了在实际编程时容易出现的问题,也包括了正文内容的深入讨论,还包括了在不同 C 编译环境中存在兼容性现象的 C 内容进行了实用而具体的指导,这部分内容不管对初学者还是对那些长期

编程的人都很有用。

每章习题中的选择题和填空题全部改编于历年全国计算机二级等级考试原题,每章的编程习题一般来源于我校 C 语言程序设计课程真实考试题,所有习题稍加修改都可作为期末试题,教师在讲完课本正文内容后,可讲解部分或全部习题(将向所有老师提供习题的解析及参考答案)。

在 C/C++ 之外的任何编程语言中,编译器都没有受到过如此之重视。这是因为 C/C++ 是一门非常复杂的语言,以至于编译器也难于构造,我们常用的编译器都不能完全符合 C/C++ 标准,以至于本书的部分测试不得不使用条件编译技术来适应不同 C++ 编译器,如下介绍一些常用的优秀 C++ 编译器。

(1) Visual C++ 6.0 编译器。它是由微软公司开发,特点是集成开发环境用户界面友好,信息提示准确,调试方便,对模板支持最完善,对硬件环境要求低,现在使用最广泛,但对标准 C/C++ 兼容较差。

(2) GCC 编译器。它是著名的开源 C++ 编译器,是类 UNIX 操作系统(例如 Linux)下编写 C++ 程序的首选,有非常好的可移植性,可以在非常广泛的平台上使用,也是编写跨平台、嵌入式程序很好的选择。GCC 3.3 与标准 C/C++ 兼容性较好。现已有一些移植在 Windows 环境下使用 GCC 编译器的 IDE(集成开发环境),例如 Dev-C++ 与 MinGW Developer Studio,其中 Dev-C++ 是能够让 GCC 在 Windows 下运行的集成开发环境。它提供了与专业 IDE 相媲美的语法高亮、代码提示,调试等功能;MinGW Developer Studio 是跨平台下的 GCC 集成开发环境。目前支持 Windows、Linux 和 FreeBSD;根据作者的实际使用,感觉使用 GCC 编译器的 IDE 错误信息提示的智能较低,错误提示不太准确,还有就是对模板支持较差,对语法检查较严格,在 Visual C++ 6.0 等编译器中编译通过的程序可能在 GCC 编译器的 IDE 还会显示有错误信息。

本书所有算法都同时在 Visual C++ 6.0、Dev-C++ 和 MinGW Developer Studio 中通过测试。读者可根据实际情况选择适当的编译器,建议选择 Visual C++ 6.0。

教师可采取多种方式来使用本书作为讲授 C 语言程序设计,应该根据学生的背景知识以及课程的学时数来进行内容的取舍。为满足不同层次的教学需求,本教材使用了分层的思想,分层方法如下:没有加星号(*)及双星号(**)的部分是基本内容,适合所有读者学习;加有星号(*)的部分适合计算机专业的读者深入学习的选学部分;加有双星号(**)的部分适合于感兴趣的同学研究。

作者为本书提供了全面的教学支持,如果在教学或学习过程中发现与本书有关的任何问题都可以与作者联系:youhongyue168@gmail.com,作者将尽力满足各位的要求,并可能将解答公布在作者的教学网站 <http://cs.scu.edu.cn/~youhongyue> 上。在教学网站上还将提供如下内容。

(1) 提供书中所有例题在 Visual C++ 6.0、Dev-C++ 和 MinGW Developer Studio 开发环境中的测试程序,今后还会提供当时流行的 C++ 开发环境的测试程序,每种开发环境还将提供基本开发过程的文档;还提供本书作者开发的软件包。

(2) 提供教学用 PowerPoint 幻灯片 ppt 课件。

(3) 向教师提供所有习题参考答案,对学生来讲,将在每学期期末(第 15 周~第 20

周)公布解压码。

(4) 高级语言程序设计问答专栏。

(5) 提供至少 6 套 C 语言程序设计模拟试题及其解答,以供学生期末复习,也可供教师出考题时参考。

(6) 提供 C++ 面向对象程序设计相关的其他资料(例如 Dev-C++ 与 MinGW Developer Studio 软件,流行免费 C/C++ 编译器的下载网址)。

希望各位读者能够抽出宝贵的时间将对本教材的建议或意见,当然也可以发表对国内外的 C 语言程序设计教学的任何意见寄给作者,这一意见将是我们再版修订教材的重要参考。

本书第 1 章~第 3 章由许春编写,第 4 章、第 6 章、第 7 章由谭斌编写,第 5 章、第 8 章、第 9 章由游洪跃编写,附录由游洪跃编写,全书由游洪跃统稿。李培宇、彭骏、何凯霖、姜琳、聂清彬、黄维、邹昌文、王文昌、周焯华、胡开文、沈洁、周德华、欧阳、文涛、文芝明和文波等人对本书做了大量的工作,参与了部分内容的编写,提供资料,调试算法;作者还要感谢为本书提供直接或间接帮助的每一位朋友,由于你们热情的帮助或鼓励激发了作者写好本书的信心以及写作热情。

本书的出版要感谢清华大学出版社各位编辑及评审专家,由于他们为本书的出版倾注了大量热情,也由于他们具有前瞻性的眼光才让读者有机会看到本书。

尽管作者有严谨的治学态度,并作了最大努力,但由于作者水平有限,书中难免有不妥之处,因此,敬请各位读者不吝赐教,以便作者有一个提高的机会,并在再版时尽力采用所提意见,尽快提高本书的水平。

作 者

2011 年 3 月

目 录

第 1 章 C 语言程序设计基础	1
1.1 C 语言的发展和主要特点	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的主要特点	1
1.2 第一个 C 语言程序以及 C 语言程序开发过程	2
1.2.1 第一个 C 语言程序	2
1.2.2 C 语言程序开发过程	5
1.3 相关知识	6
1.3.1 算法	6
1.3.2 数据结构的基本概念	6
1.3.3 计数制及十进制与其他计数制之间的转换	7
1.4 程序陷阱	8
1.5 习题	8
第 2 章 数据类型及其运算	10
2.1 标识符与关键字	10
2.2 C 语言的数据类型	11
2.3 常量与变量	14
2.3.1 整型量	15
2.3.2 实型量	17
2.3.3 字符型量	18
2.3.4 符号常量	20
2.3.5 类型转换	21
2.4 基本运算符和表达式	23
2.4.1 运算符的种类、优先级和结合性概述	23
2.4.2 优先级和结合性	24
2.5 相关知识：程序和程序设计	30
2.5.1 程序	30

2.5.2 程序设计	30
2.6 程序陷阱	30
2.7 习题	31
第3章 C语言程序结构及相关语句	35
3.1 相关知识	35
3.1.1 算法描述方法	35
3.1.2 结构化程序	35
3.1.3 模块化结构	36
3.2 顺序语句	36
3.2.1 C程序的语句	36
3.2.2 数据输出语句	37
3.2.3 数据输入语句	42
3.3 分支结构程序	46
3.3.1 关系运算符和表达式	46
3.3.2 逻辑运算符和表达式	47
3.3.3 if 语句	49
3.3.4 条件运算符和条件表达式	56
3.3.5 switch 语句	57
3.3.6 程序举例	60
3.4 循环结构程序	62
3.4.1 while 语句	62
3.4.2 do...while 语句	64
3.4.3 for 语句	65
3.4.4 转移语句	68
3.4.5 程序举例	71
3.5 程序陷阱	73
3.6 习题	75
第4章 函数	83
4.1 函数的定义与调用	83
4.1.1 函数定义的一般形式	83
4.1.2 函数调用的一般形式	85
4.1.3 函数的声明和函数原型	86
4.2 函数的参数和函数的值	88
4.2.1 函数的参数	88
4.2.2 函数的值	89
4.3 函数的嵌套调用	90

4.4	递归函数	91
4.5	变量的作用域	93
4.5.1	局部变量	93
4.5.2	全局变量	95
* 4.6	变量的存储类型和生存期	98
4.6.1	自动变量	98
4.6.2	外部变量	101
4.6.3	静态变量	101
4.6.4	寄存器变量	103
* 4.7	内部函数和外部函数	104
4.7.1	内部函数	104
4.7.2	外部函数	104
4.8	相关知识	105
4.8.1	程序设计方法和风格	105
4.8.2	结构化程序设计	105
* 4.8.3	面向对象的程序设计	105
4.9	实例研究: 汉诺塔问题	106
4.10	程序陷阱	107
4.11	习题	109
第 5 章	数组和指针	115
5.1	一维数组的定义和引用	115
5.1.1	定义一维数组	115
5.1.2	引用数组元素	116
5.2	二维数组	119
5.2.1	定义二维数组	119
5.2.2	引用二维数组元素	119
5.2.3	二维数组的初始化	121
5.3	数组名作为函数参数	123
5.3.1	数组元素作函数参数	123
5.3.2	数组名作为函数参数	124
* 5.3.3	多维数组作函数的参数	126
5.4	字符数组与字符串	127
5.4.1	字符数组	127
5.4.2	字符串	129
5.4.3	字符串常用函数	130
5.5	数组程序举例	134
5.6	指针变量的声明和初始化	137

5.7	指针运算符	138
5.8	指向 void 的指针	139
5.9	函数参数的引用传递	139
5.10	指针变量和数组	142
5.10.1	指针变量与一维数组	142
* 5.10.2	指针变量与二维数组	145
5.11	字符指针、字符数组和字符串	150
5.11.1	字符指针与字符数组的区别	150
5.11.2	字符指针数组和字符串数组	151
* 5.11.3	指向字符指针的指针	153
5.12	相关知识	155
5.12.1	查找技术	155
5.12.2	排序技术	155
5.13	实例研究	156
5.13.1	顺序查找	156
5.13.2	冒泡排序	157
5.14	程序陷阱	158
5.15	习题	158
第 6 章	用户定制数据类型及位运算	165
6.1	结构	165
6.1.1	概述	165
6.1.2	结构的声明	165
6.1.3	结构类型变量的引用	168
6.1.4	结构变量的初始化	169
6.1.5	结构数组	170
6.1.6	指向结构类型的指针	175
6.2	联合	178
6.2.1	联合的概念	178
6.2.2	联合的定义	178
6.2.3	联合变量的说明	179
6.2.4	联合变量的赋值和使用	180
6.3	枚举类型	181
6.4	类型声明: typedef	186
6.5	位运算符	187
6.5.1	位运算符介绍	188
* 6.5.2	位运算综合举例	192

* 6.6	位段结构	195
6.6.1	位段结构的定义和位段结构变量的说明	195
6.6.2	位段结构的使用	197
* 6.7	相关知识	200
6.7.1	线性表及其顺序存储结构	200
6.7.2	栈和队列	200
** 6.8	实例研究：顺序栈	201
6.9	程序陷阱	205
6.10	习题	206
第 7 章	预处理命令	210
7.1	概述	210
7.2	文件包含	210
7.3	宏定义	211
7.3.1	无参宏定义	211
7.3.2	带参宏定义	213
* 7.3.3	取消宏 #undef	219
* 7.4	条件编译	219
** 7.5	宏 assert(断言)	223
* 7.6	相关知识：软件工程基础	223
7.7	程序陷阱	224
7.8	习题	225
第 8 章	文件	228
8.1	文件概念	228
8.2	文件指针	229
8.3	文件的打开与关闭	229
8.3.1	文件打开的函数 fopen()	229
8.3.2	文件关闭的函数 fclose()	231
8.4	文件检测函数	231
8.5	文件操作函数	232
8.5.1	字符读写函数：fgetc()/fputc()	232
8.5.2	字符串读写函数：fgets()/fputs()	235
8.5.3	数据块读写函数：fread()与fwrite()	238
8.5.4	格式化读写函数：fscanf()/fprintf()	242
8.6	随机读写文件	245
* 8.7	相关知识	248
8.7.1	数据库系统的基本概念	248

8.7.2	数据库设计与管理	249
8.7.3	数据库的物理设计	249
8.7.4	数据库管理	249
**8.8	实例研究: 人事管理系统	249
8.8.1	需求分析	249
8.8.2	功能描述	250
8.8.3	总体设计	250
8.8.4	系统实现	252
8.9	程序陷阱	259
8.10	习题	260
第9章	高级主题	263
**9.1	变长参数表	263
**9.2	命令行参数	265
9.3	动态内存分配与释放	267
9.3.1	动态内存分配函数: malloc()	267
9.3.2	动态内存释放函数: free()	267
9.3.3	动态内存处理实例: 线性链表	268
9.4	指针的深入讨论	271
**9.4.1	指向函数的指针变量	271
*9.4.2	返回指针的函数	272
**9.5	相关知识: 线性链表	273
9.6	程序陷阱	274
9.7	习题	275
附录A	常用C编译器使用方法	278
A.1	Visual C++ 6.0	278
A.2	MinGW Developer Studio	280
A.3	Dev-C++	283
附录B	常用字符ASCII码对照表	286
附录C	C运算符与优先级	287
附录D	C常用库函数	288
参考文献		294

第 1 章 C 语言程序设计基础

1.1 C 语言的发展和主要特点

1.1.1 C 语言的发展

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,同年由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《The C Programming Language》一书,在此书的附录中提供了 C 语言参考手册,在此后的几年中,该书一直被广泛作为 C 语言的实现规范,成为事实上的非官方 C 语言标准,一般将此书提出的 C 语言标准简称为 K&R C。

C 语言在随后的几年中得到了快速发展,实际使用的 C 语言已远远超越了 K&R C 所描述的范围,同时 K&R C 在许多细节上也没有达到足够精确的要求,迫切要求进一步标准化 C 语言,在 1983 年,美国标准化组织(American National Standard Organization, ANSI)成立了一个 C 语言工作小组,开始了进行 C 语言标准化工作,此小组确认了 C 语言的常用特性,并对 C 语言本身做了一些改进,同时引入了 C 语言的一些新特征。1989 年,ANSI 发布了新的 C 语言标准——ANSI C,简称为 C89。此后,该标准被国际标准化组织(International Standard Organization, ISO)修改后作为 ISO C 标准。

当前广泛使用的 C 编译环境都实现了 ANSI C,不同版本的 C 编译环境只略有差异。

在 ANSI 标准化后,C 语言的标准化在相当长一段时间内都保持不变,直到 1999 年,ISO 加入了一些新特性,发布了 ISO 8999:1999,此版本通常称为 C99,但是各公司对 C99 普遍缺乏兴趣,大部分公司在最新的 C 编译器中都不支持 C99 的新特征。

为了读者使用,本书介绍的程序都符合 ANSI C 标准,并同时在 Visual C++ 6.0 和 MinGW Developer Studio 和 Dev-C++ 开发环境中进行了严格的测试。

1.1.2 C 语言的主要特点

C 语言是最受欢迎的计算机语言,C 语言既可以用来编写应用程序,也可以用来编写系统程序,具体来讲,C 语言具有如下主要特点。

1. 使用方便,功能强大

C 语言只定义了 32 个关键字,易于初学者记忆与掌握,C 语言程序书写形式自由,并且含有丰富的操作和数据类型,允许用户定义数据类型,便于处理各种复杂数据和实现

各种复杂功能。

2. 便于模块化编程

C语言是一种函数式语言,使用函数作为程序的基础模块,使C语言程序设计十分方便,C语言程序层次分明且便于维护。

3. C程序可移植性好

按照ANSI C设计的程序,可以不做修改就能运用于各种不同计算机与不同操作系统下,使得C语言适合于开发跨平台的程序开发。

1.2 第一个C语言程序以及C语言程序开发过程

1.2.1 第一个C语言程序

C语言程序的结构十分严谨,下面介绍著名的“Hello, world!”程序,此程序经常被用于介绍各种语言的第一个程序,其功能是在屏幕上输出字符串“Hello, world!”。

例 1.1 在屏幕上输出“Hello, world!”。

```
/* 文件路径名:el_1\main.c */
#include <stdio.h>           /* 包含库函数 printf()所需要的信息 */
#include <stdlib.h>          /* 包含库函数 system()所需要的信息 */

int main(void)               /* 主函数 main() */
{
    printf("Hello, world!\n"); /* 输出"Hello, world" */

    system("PAUSE");          /* 调用库函数 system(),输出系统提示信息 */
    return 0;                 /* 返回值 0, 返回操作系统 */
}
```

运行上面程序时,将在屏幕上输出如下信息:

```
Hello, world!
请按任意键继续 ...
```

为了使读者更好地理解,下面将详细地剖析上面的程序。

1. 注释

上面程序的第一行如下:

```
/* 文件路径名:el_1\main.cpp */
```

这行不是程序代码,它只是注释,告诉读者程序的文件路径名,位于/*和*/之间的任何文本都是注释。应养成给程序添加注释的习惯,添加足够的注释,可为日后自己(和其他程序员)更加容易地理解程序的作用和工作方式。