



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计 教程

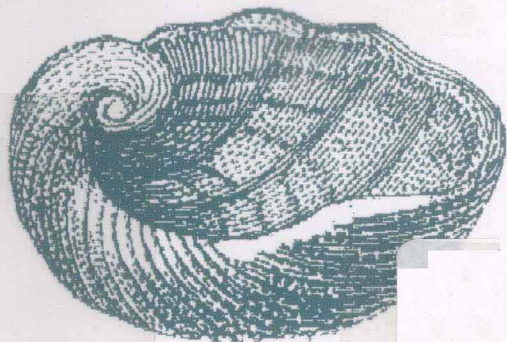
C Programming

杨治明 雷亮 主编

伍建全 焦晓军 副主编

向毅 王双明 主审

- 以案例驱动模式展开知识讲解任务，引导读者“从做中学”
- 改变传统的被动式灌输语法知识，引导读者自主学习和探索
- 针对不同应用领域延伸阅读，引导读者掌握C核心编程方法



高校系列



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计 教程

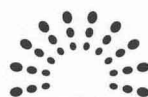
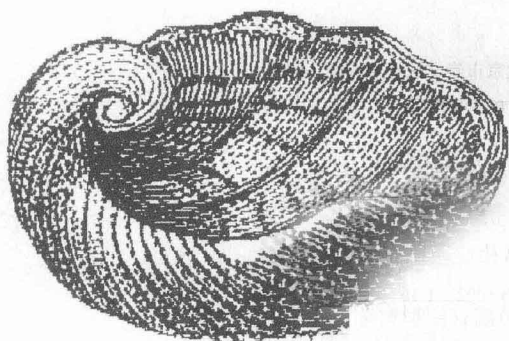
要 封 存 内

C Programming

杨治明 雷亮 主编

伍建全 焦晓军 副主编

向毅 王双明 主审



高校系列

人民邮电出版社

北 京

图书在版编目(CIP)数据

C语言程序设计教程 / 杨治明, 雷亮主编. — 北京:
人民邮电出版社, 2012.3
21世纪高等学校计算机规划教材
ISBN 978-7-115-27087-0

I. ①C… II. ①杨… ②雷… III. ①
C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(2012)第013900号

内 容 提 要

本书定位于将C语言作为计算机编程入门语言,以帮助读者树立计算机程序设计的思想,培养学生程序设计基本能力为目标的教材。

全书以程序设计为主线,采用案例驱动模式展开讲解,通过案例和问题引入主要教学内容,重点讲解程序设计的思想和方法,并结合相关的语法知识。全书主要包括3方面的内容:基本内容(数据表达、数据处理和流程控制)、常用算法和程序设计风格以及C语言应用中的一些处理机制(编译预处理和命令行参数等)。涉及数据类型、表达式、分支、循环、函数、数组、指针、结构、文件的概念、应用以及指针和各种构造类型的混合运用、基本算法等内容。

本书可以作为高等院校和计算机等级考试的教学用书,也可作为对C语言程序设计感兴趣的读者的自学用书。

工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

C语言程序设计教程

- ◆ 主 编 杨治明 雷 亮
副 主 编 伍建全 焦晓军
主 审 向 毅 王双明
责任编辑 蒋 亮
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 19 2012年3月第1版
字数: 501千字 2012年3月北京第1次印刷

ISBN 978-7-115-27087-0

定价: 39.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

作为一种计算机程序设计语言，C 语言功能丰富，表达能力强，使用灵活方便，应用广泛，可移植性好。它具有高级语言形式、低级语言功能，集高级语言和低级语言的优点于一身，是计算机应用领域最有影响的计算机程序设计语言之一，因而被大多数高等学校采用，作为典型的计算机语言教学课程，也被选为计算机等级考试、全国计算机应用证书考试等多种计算机技能考试的考试内容。此外，C 语言作为一门实用且功能强大的程序设计语言，被程序设计人员广泛使用。因此，C 语言是一门十分重要的程序设计语言。

学习程序设计语言的目的在于应用程序设计语言编写程序，解决学习和工作中遇到的实际问题，而不仅仅是掌握该语言的语法。读者在初学 C 语言程序设计时，往往容易为该语言的语法规则所羁绊，接触到的大量语法知识难以记忆，难以理解，更不会掌握应用，难免产生为难情绪，容易失去学习 C 语言的信心，更找不到学习的兴趣。作者根据多年从事 C 语言教学的经验，结合应用型本科人才培养的需求和教育教学改革的成果，对课程的教学理念和教学方法进行了认真思考，组织课程组同事共同完成了本书的编写。

本书具有以下特色。

(1) 以案例驱动模式展开知识讲解任务。以问题驱动的方式，通过案例描述、思路分析、程序实现、知识聚焦的方式将程序设计的基本概念、基本方法逐步展开，把重点放在解题思路，引导读者“从做中学”。

(2) 从开始学习就使读者将注意力集中在如何应用计算机编程来解决问题上，从具体案例的解决方法来理解 C 语言程序设计的特点和程序总体框架，通过案例本身学习某一类问题的解决方法和算法设计，以程序编写带动语法规则的教学。

(3) “学以致用”是 C 语言教学的基本理念，为帮助读者掌握 C 语言程序设计的知识，具备应用 C 语言开发应用软件的能力，提高计算机应用水平，本书最后安排一章课程设计要求，根据学生和读者的特点，设计了几个趣味性较强的课程设计要求并给出了参考源代码，以帮助读者理解和掌握 C 语言的基本知识，提高学习、探究 C 语言实际问题功能的兴趣。

(4) 改变传统的被动式灌输语法知识，引导读者自主学习和探索，使读者不再陷入语言的语法规则泥潭，而重在程序设计方法的学习和算法的探究，通过程序设计实践来掌握语言。

(5) 针对不同应用领域读者的不同需求，在个别章节安排了延伸阅读内容，以满足特殊需求的读者深入学习特定知识点的需求。

(6) 引导读者掌握 C 语言的核心编程方法，提高应用能力。引导学生在解题编程的实践中探索其中的规律，将感性认识升华到理性高度。

全书分为理论部分和实验部分。理论部分共分 10 章。第 1 章介绍 C 语言的基本结构、特点、发展历史、程序调试方法，简单 C 语言程序的设计等；第 2 章介

绍 C 语言程序的基本数据类型和输入/输出,是 C 语言的基本语法部分;第 3 章、第 4 章、第 5 章以案例为驱动,以大量实际问题的分析和解决方法为例,循序渐进地介绍结构化程序设计的编程方法,包括应用循环、分支及函数的程序设计思想和方法;第 6 章、第 7 章、第 8 章介绍如何编写具有构造数据类型的程序;第 9 章介绍磁盘数据存储程序的设计方法;第 10 章是全书各章节知识点的综合应用,通过一个综合性的案例,介绍应用软件的开发方法,并提供了几个趣味性的题目,供学生练习。

本书的编写工作由杨治明主持。第 1 章和第 9 章及相关实验部分由杨治明编写,第 2 章、第 4 章、第 10 章、附录及相关实验部分由雷亮编写,第 3 章、第 5 章、第 8 章及相关实验部分由焦晓军编写,第 6 章和第 7 章及相关实验部分由伍建全编写。全书由杨治明统稿。向毅博士在百忙之中审阅了全书理论部分,王双明老师审阅了实验部分,陈应祖老师对本书的编写提出了宝贵意见。

本书的编写参考了近年来出版的大量书籍及相关技术资料,吸取了许多同仁和专家的宝贵经验,在此一并表示衷心的感谢!尽管我们付出了很大的努力,但由于水平有限,书中难免出现错误或不妥之处,恳请读者批评指正。

编者

2011 年 11 月

目 录

第 1 章 C 语言概述.....1

1.1 简单的 C 语言程序.....1

1.2 C 语言的发展历史与特点.....3

1.2.1 C 语言的发展历史.....3

1.2.2 C 语言的特点.....4

1.3 C 程序的开发步骤和上机调试流程.....5

1.3.1 C 程序的开发步骤.....5

1.3.2 C 程序的上机开发过程.....6

习题 112

第 2 章 基本数据类型的输入/输出.....13

2.1 C 语言的输入/输出.....13

2.2 整型数据的输入/输出.....13

2.3 浮点型数据的输入/输出.....14

2.4 字符数据的输入/输出.....15

2.5 字符串数据的输入/输出.....15

2.6 格式化输入/输出举例.....17

2.7 阅读材料.....18

2.7.1 格式化输出函数 printf 的格式说明 和使用.....18

2.7.2 格式化输入函数 scanf 的格式说明 和使用.....19

习题 221

第 3 章 流程控制.....22

3.1 算法.....22

3.1.1 算法的概念.....22

3.1.2 算法的表达方式.....23

3.1.3 基本流程控制结构.....25

3.1.4 案例 3.1 求 $1+1/2+\cdots+1/100$ 的和.....26

3.2 选择语句.....26

3.2.1 案例 3.2 求三个整数的最大值 (if 语句).....26

3.2.2 案例 3.3 百分制成绩转换成五分 制成绩 (多分支 if 语句).....29

3.2.3 案例 3.4 判断所输入的一个字符是 数字、空白符还是其他字符 (switch 语句).....31

3.2.4 案例 3.5 百分制成绩转换成五分 制成绩 (switch 语句).....33

3.3 循环语句.....34

3.3.1 案例 3.6 求 $1+1/2+\cdots+1/100$ 的和 (while 语句).....34

3.3.2 案例 3.7 求 $1+1/2+\cdots+1/100$ 的和 (do...while 语句).....35

3.3.3 案例 3.8 求 $1+1/2+\cdots+1/100$ 的和 (for 语句).....37

3.4 转向语句.....39

3.4.1 案例 3.9 判断所输入的一个大于 1 的 正整数是否是素数 (break 语句).....39

3.4.2 案例 3.10 输出 100~200 之间能 被 3 整除的数 (continue 语句).....41

3.5 应用举例.....42

3.5.1 案例 3.11 计算 $1!+2!+\cdots+10!$ (并讨论溢出问题).....42

3.5.2 案例 3.12 计算级数 $1-1/3+1/5-1/7+\cdots$ 的和.....44

3.5.3 案例 3.13 统计输入的数字字符、字母 字符和其他字符的个数.....45

3.5.4 案例 3.14 求两个正整数的最大公约 数和最小公倍数.....45

3.5.5 案例 3.15 将一个正整数逆序输出.....46

3.5.6 案例 3.16 输入日期并检查其合理性, 直到输入合理为止.....47

3.6 阅读材料.....48

3.6.1 C 语言的语句.....48

3.6.2 goto 语句简介.....49

3.6.3 exit() 函数.....49

3.6.4 程序调试简介.....50

习题 351

第 4 章 函数.....54

4.1 模块化程序设计.....54

4.1.1 模块化程序设计的基本思想.....54

4.1.2 C 语言程序设计的基本结构.....55

4.2 函数定义和调用.....55

4.2.1 案例 4.1 求 3 个整数的最大值
(函数定义和调用).....574.2.2 案例 4.2 验证哥德巴赫猜想
(在 1000 以内验证).....574.2.3 案例 4.3 计算 $1^k+2^k+\cdots+n^k$ (函数
原型、函数的嵌套调用).....59

4.3 标识符的作用域.....60

4.3.1 案例 4.4 块作用域.....60

4.3.2 案例 4.5 文件作用域.....61

4.3.3 函数原型作用域.....61

4.3.4 函数作用域.....62

4.4 变量的存储种类.....62

4.4.1 自动变量.....62

4.4.2 寄存器变量.....62

4.4.3 案例 4.6 外部变量.....63

4.4.4 案例 4.7 静态变量.....64

4.5 指针基础.....64

4.5.1 地址和指针的概念.....64

4.5.2 指针变量.....65

4.5.3 案例 4.8 两数交换(指针做参数).....65

4.5.4 案例 4.9 输入 10 个整数求它们的最大
值、最小值和总和(指针做参数).....664.5.5 案例 4.10 统计百分制成绩各
分数段的人数(指针做返回值).....67

4.6 递归函数.....68

4.6.1 案例 4.11 计算 $4!$ (递归函数的
定义和递归调用过程分析).....68

4.6.2 案例 4.12 计算整数各位数字之和.....69

4.6.3 案例 4.13 计算 $\sqrt{a}$70

4.6.4 案例 4.14 汉诺塔问题.....70

习题 4.....71

第 5 章 编译预处理.....74

5.1 宏.....74

5.1.1 案例 5.1 无参宏.....75

5.1.2 案例 5.2 有参宏.....75

5.2 文件包含.....76

5.3 条件编译.....77

5.3.1 条件编译命令.....77

5.3.2 案例 5.3 头文件设计.....78

5.4 阅读材料.....80

5.4.1 程序的多文件组织方法.....80

5.4.2 多文件程序的编译和链接.....80

习题 5.....81

第 6 章 数组.....83

6.1 一维数组.....83

6.1.1 一维数组的定义、初始化和使用.....83

6.1.2 最大值、最小值和平均值.....86

6.1.3 一维数组作为函数的参数.....88

6.1.4 数组排序.....90

6.1.5 数组查找.....95

6.2 二维数组.....98

6.2.1 二维数组的定义、初始化和使用.....98

6.2.2 矩阵转置.....100

6.2.3 二维数组的本质.....101

6.3 字符串.....104

6.3.1 字符数组和字符串.....104

6.3.2 字符串的输入和输出.....105

6.3.3 字符串处理.....107

习题 6.....111

第 7 章 指针.....120

7.1 简介.....120

7.2 什么是指针.....120

7.2.1 变量、变量的值与变量的地址.....120

7.2.2 指针.....122

7.2.3 定义指针变量.....123

7.2.4 间接运算符.....123

7.2.5 指针赋值.....125

7.2.6 指针作为函数的参数.....126

7.2.7 返回指针的函数.....130

7.3 指针与一维数组.....131

7.3.1 使指针指向数组.....131

7.3.2 指针算术运算	131	8.2.1 案例 8.3 单向链表的建立和 基本操作	186
7.3.3 指针关系运算	138	8.2.2 案例 8.4 猴子选大王	189
7.3.4 一维数组名与指针	140	8.3 共用体	192
7.3.5 向函数传递数组实际上是 传递指针	142	8.3.1 案例 8.5 共用体类型及其变量的 定义和使用	192
7.4 指针与二维数组	145	8.3.2 案例 8.6 学生、教职工基本信息 管理	193
7.4.1 二维数组的本质	145	8.4 枚举	200
7.4.2 指向数组的指针与二维数组名	146	8.4.1 案例 8.7 枚举类型及其变量的 定义和使用	201
7.4.3 用指向元素的指针访问二维 数组	151	8.4.2 案例 8.8 枚举类型数据的输入和 输出	201
7.5 指针数组	153	8.5 自定义数据类型举例	202
7.6 带参数的 main 函数	154	8.6 阅读材料	204
7.7 指向指针的指针	155	8.6.1 位域	204
7.8 指向函数的指针	156	8.6.2 无名共用体类型的使用	205
7.8.1 指向函数的指针的定义和使用 方法	156	8.6.3 类型别名	206
7.8.2 指向函数的指针作函数参数	157	习题 8	206
7.9 字符串处理函数	159	第 9 章 文件	209
7.9.1 字符串的输入输出 (gets 和 puts)	159	9.2 文件概述	209
7.9.2 求字符串长度 (strlen)	160	9.1.1 文件的概念	210
7.9.3 字符串复制 (strcpy、strncpy) 和 字符串连接 (strcat)	160	9.1.2 文本文件与二进制文件	210
7.9.4 字符串比较 (strcmp)	163	9.1.3 缓冲文件系统	211
7.9.5 字符串中字符的大小写转换 (strlwr 和strupr)	164	9.1.4 文件类型指针	211
7.10 动态内存分配	164	9.2 文件的打开与关闭	212
7.10.1 C 语言的程序内存空间	165	9.2.1 打开文件 (fopen() 函数)	212
7.10.2 用 malloc 函数分配一块内存	165	9.2.2 关闭文件函数 (fclose 函数)	214
7.10.3 用 calloc 函数分配连续的内存 空间	167	9.3 文本文件的读写	214
7.10.4 用 realloc 函数改变内存块的 大小	168	9.3.1 案例 9.1 复制文件 (fputc 函数和 fgetc 函数)	214
习题 7	170	9.3.2 案例 9.2 复制文件 (fgets 函数和 fputs 函数)	216
第 8 章 自定义数据类型	178	9.3.3 案例 9.3 学生基本信息的顺序读/ 写 (fprintf 函数和 fscanf 函数)	218
8.1 结构体	178	9.4 二进制文件的读写	220
8.1.1 案例 8.1 复数的四则运算	183	9.4.1 案例 9.4 学生基本信息的顺序读/ 写 (fread 函数和 fwrite 函数)	220
8.1.2 案例 8.2 学生成绩统计	184	9.4.2 案例 9.5 学生基本信息的随机读/	
8.2 单向链表	186		

写 (fseek 函数)	222	实验九 指针与数组	250
9.5 阅读材料	224	实验十 指针与函数	251
9.5.1 ferror() 函数	224	实验十一 结构体与单向链表	253
9.5.2 clearerr() 函数	224	实验十二 文件	255
9.5.3 feof() 函数	224	附录 A 常用字符与 ASCII 码 对照表	257
习题 9	224	附录 B C 语言的 32 个关键字	259
第 10 章 课程设计	226	附录 C C 语言常用标准库函数	260
10.1 课程设计的目的和要求	226	附录 D Turbo C 2.0 程序调试 技术	266
10.2 课程设计举例——学生成绩管理 系统	226	附录 E Visual C++ 6.0 程序调试 技术*	270
10.3 课程设计选题	233	附录 F 推箱子游戏源代码	272
实验一 C 语言编译环境与程序 开发	235	附录 G 贪吃蛇游戏源代码	275
实验二 输入/输出程序设计	236	附录 H 扫地雷游戏源代码	279
实验三 选择结构	238	附录 I 速算 24 源代码	288
实验四 循环结构	241		
实验五 函数	244		
实验六 递归	245		
实验七 数组的使用	246		
实验八 字符串处理	248		

第 1 章

C 语言概述

教学目标

1. 了解 C 语言的发展历史。
2. 理解 C 语言的特点。
3. 掌握 C 程序的开发步骤和上机调试流程。

1.1 简单的 C 语言程序

随着计算机技术的发展和数据处理技术的广泛应用，掌握一门计算机程序设计语言已经成为当代大学生必备的基本素质之一。作为应用最广泛的程序设计语言，C 语言因其良好的兼容性和可移植性，被广泛应用于系统软件和应用软件的开发。

如何尽快学会用 C 语言进行程序设计是初学者最为关心的事情。学习一门计算机语言与学习一种人类语言一样，需要不断地练习和反复地实践，通过一系列的编程实践，逐步掌握程序设计语言的语法规则，领会程序设计的基本思想和方法。任何熟练的程序员，其编程技能和经验都是通过程序设计实践逐渐积累起来的。

作为程序设计语言的初学者，往往缺乏对计算机运行的机制、原理的了解和程序设计语言的相关知识、编程经验，需要从模仿简单的程序开始，逐步体会程序设计语言的语法规则和编程思想、方法。

为了让读者体会对 C 语言程序设计有一个初步的感性认识，首先让我们从最简单的 C 程序开始。

案例 1.1 在屏幕上打印出如下内容

Hello, World!

◆ 问题描述

我们希望通过运行一个程序在屏幕上打印出“Hello, World !”这句短语。

◆ 程序实现

```
/*本程序实现在终端上输出短语“Hello World !”*/ /*注释语句*/  
#include <stdio.h> /*文件包含语句*/  
int main(void) /*定义主函数 main( ) */
```

```

{
printf(" Hello,World !\n ");    /*调用 printf( ) 函数输出文字*/
return 0;                      /*main( ) 函数的返回值*/
}

```

◆ 知识聚焦

(1) 注释语句是用来方便读者理解程序功能的注解，只要方便读者理解，在 C 程序中可以将注释语句置于程序任何位置，但注释内容必须包含在 “/*” 和 “*/” 之间，否则会被认为是程序。

(2) 文件包含语句也称为编译预处理命令，用于说明其后调用的函数包含在哪个库文件中。本例中 `stdio.h` 是系统提供的一个文本文件（我们常常称后缀名位 `.h` 的文件为头文件），它表示其后调用的函数可能会由标准输入/输出库函数提供。



注意

编译预处理语句末不加分号。

(3) `int main( )` 语句定义了一个名为 `main` 的主函数，该函数的返回值是整数类型（`int`），函数无需任何参数。

紧跟函数名后的括号内标明 `void` 或不加任何内容表明该函数不需要任何参数，如需要参数，C 语言规定必须在函数名后面的括号中标明参数的数据类型、参数名称的信息，相关知识读者可参阅本书第 4 章。

在 C 语言中，`main( )` 函数是一个特殊的函数，称为“主函数”。任何 C 程序都必须有且只有一个主函数。当程序运行时，首先从 `main( )` 函数开始执行，而不管该函数在源程序的什么位置。

(4) 由一对 “{ }” 括起来的众多语句构成了函数体，用于实现函数的功能。

(5) `printf( " Hello, World !\n " );` 这是一句完整的 C 语句，它的功能是把引号内的内容原样输出到终端上。

这个语句由两部分组成：函数调用和分号。

`printf( )` 是系统提供的标准输出函数，括号内的内容 “`Hello, World !\n`” 是函数的参数，其中 `\n` 仅代表一个字符，叫转义字符，表示回车换行；“`;`”（分号）表示语句的结束。

(6) 最后一句是函数返回，作用是结束 `main( )` 函数的运行，并向操作系统返回一个整数 0。由于 `main( )` 函数定义为整型，故返回值为任何整数都应该是正确的，但是根据程序员的习惯，返回值为 0，表示程序运行正常，而其他数字标明程序错误的类型，方便程序员查找程序设计的错误。

(7) C 语言是大小写敏感的，本程序中各语句涉及的如 `main`、`printf`、`int` 等都是小写英文字符，如将其中的任何一个字符改成大写将造成编译错误。例如，`sum` 和 `Sum` 将被认为是两个不同的标识符。

◆ 问题思考

(1) 若要输出短语 “I love C language”，该如何编写程序？

(2) 下面的语句运行结果是什么？

```
printf("C Language is fun\n I love C language!\n");
```

案例 1.2 计算两个整数的和

◆ 问题描述

我们利用计算机程序实现计算两个整数的和，并输出计算结果。

◆ 程序实现

```
#include <stdio.h>
int main(void)
{
    int a,b,sum;           /*声明 3 个整数类型的变量*/
    a=3;                   /*将整数 3 赋予变量 a*/
    b=5;                   /*将整数 5 赋予变量 b*/
    sum=a+b;               /*计算 a 与 b 的和，并将和赋予 sum*/
    printf(" sum is :%d\n ",sum); /*调用 printf( ) 函数输出结果*/
    return 0;
}
```

◆ 知识聚焦

(1) 变量声明用于界定变量的数据类型，实质是计算机系统中申请一定大小的内存空间用于保存数据。本例中将 3 个变量都定为整型，那么他们只能用于保存整数，其他类型的数值（如小数）就不能存放在该变量中。

(2) printf() 函数与上一案例调用时相比有 2 个不同之处。

① 括号内包含有 2 个参数，“sum is :%d\n”和 sum，参数间用“,”分开。

② 引号内出现%d 符号，它标明该处将打印出一个整型数字，该数字大小等于 sum 的值。在文件 stdio.h 中，printf() 函数的原型为：

```
int printf(const char *format,...);
```

◆ 问题思考

(1) 如果要计算 3 个数的和，案例 1.2 的程序该如何修改？

(2) 如果输出时要求列出代数式，案例 1.2 的程序该如何修改？

1.2 C 语言的发展历史与特点

1.2.1 C 语言的发展历史

计算机程序是由一系列计算机可以识别的指令组成的代码集合，这些代码是人们为了解决某种特定的问题而按照一定的规则专门为计算机编排的执行步骤，由计算机严格按步骤去执行。众所周知，计算机能够识别的指令为由 0 和 1 组成的二进制代码，但人工设计二进制的代码非常困难、容易出错又非常不适合人阅读和修改，因此人们设计了适合人阅读的程序设计语言。

程序设计语言有高级语言和低级语言之别。低级语言分机器语言（二进制语言）和汇编语言（符号语言），这两种语言都是面向硬件的语言，和具体硬件的指令系统密切相关。机器语言用指

令代码编写程序，而符号语言用指令助记符来编写程序。

高级语言与计算机的硬件结构及指令系统无关，它有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且容易学习掌握。但高级语言编译生成的程序代码一般比用汇编程序语言设计的程序代码要长，执行的速度也慢。所以汇编语言适合编写一些对速度和代码长度要求高的程序和直接控制硬件的程序。

C 语言是应用最广泛的高级程序设计语言之一，作为一种计算机高级语言，C 语言是由 1972 年贝尔实验室的 Dennis Ritchie 在 B 语言的基础上发展起来的。

1973 年 Dennis Ritchie 和 Ken Thompson 合作完成了当时最典型、最庞大、最复杂的分时操作系统——UNIX 系统的 C 语言改写，成功开发了 UNIX 第五版（以前版本是 Dennis Ritchie 和 Ken Thompson 用汇编语言编写的）。

此后，C 语言还经过了多次改进，逐渐形成了不依赖于具体机器的可移植的 C 语言编译软件，但主要还是在贝尔实验室内部使用，指导 UNIX 第六版发布，C 语言的优点才引起世人的普遍关注，不同的组织在不同的操作系统上也各自开发了不同版本的 C 语言编译系统。

1978 年 B.W.Kernighan 和 Dennis Ritchie（合称 K & R）合著出版了《The C Programming Language》一书，该书成为各种版本 C 语言编译系统的基础，也是当时事实上的 C 语言标准。该书也是 C 语言发展史上的一个重要的里程碑。

1983 年美国国家标准协会（American National Standards Institute ANSI）根据各种版本 C 语言编译软件对 C 语言的改进和扩充，发布了第一个 C 语言标准草案（83 ANSI C）。

1983 年夏天，ANSI 成立了 X3J11 委员会，目的是产生一个 C 标准。X3J11 在 1989 年末提出了一个他们的报告（即 89 ANSI C 89）。

1990 年，国际标准化组织 ISO（International Standards Organization）接受了 89 ANSI C 为 ISO C 的标准（ISO9899-1990）。1994 年，ISO 再次修订了 C 语言的标准。

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 编译系统所实现的语言功能和语法规则又略有差别。

1.2.2 C 语言的特点

C 语言是一种成功的系统描述语言，用 C 语言开发的 UNIX 操作系统显示了该语言具备的强大功能；同时 C 语言又是一种通用的程序设计语言，在国际上广泛流行。世界上很多著名的计算公司都成功地开发了不同版本的 C 语言，很多优秀的应用程序也是使用 C 语言开发的，它是一种很有发展前途的高级程序设计语言。

通过上节对 2 个简单 C 程序的介绍，读者对 C 程序的风格有了初步的感性认识，其实无论哪种版本的 C 语言，都有如下共同的特点：

1. C 语言是一种结构化语言

C 语言主要以函数的形式提供给用户，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。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。同时也实现了代码和数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。

2. 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，主要用小写字母表示。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，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,可用来写系统软件。所以也有把C语言称作一种中级语言。

3. 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。

4. 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。且计算功能、逻辑判断功能强大。

5. C语言适用范围大

适合于多种操作系统,如Windows、DOS、UNIX等;也适用于多种机型。C语言对编写需要硬件进行操作的场合,明显优于其他解释型高级语言,有一些大型应用软件也是用C语言编写的。C语言具有绘图能力强,良好的可移植性,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。它是数值计算的高级语言。

6. 目标代码质量高,程序执行效率高

用C语言编写的程序编译成目标代码后,一般只比用汇编语言编写的程序生成的目标代码效率低10%~20%。这方面C语言比其他高级语言的优势更为明显。

7. 用C语言编写的程序可移植性好

用C语言编写的程序基本上可以不做修改就可以应用在不同的操作系统上。

8. 程序编写自由度大,语法限制不太严格

虽然C语言也是强类型语言,但它的语法比较灵活,允许程序编写者有较大的自由度。

当然,C语言也有其自身的不足,如数据类型检查不够严格,表达式容易引起歧义,不自动检查数据越界等。这些缺点也是初学者感到困难的地方,需要读者在学习时充分注意到其不足,否则遇到相关问题,很难发现程序中出现的错误而不知道从何处入手解决。

1.3 C程序的开发步骤和上机调试流程

1.3.1 C程序的开发步骤

上节我们曾经介绍过,计算机语言有高级语言和低级语言之分,其中只有低级语言的二进制机器语言才能被计算机识别并执行,但人们编程所用的计算机语言多为高级语言(少部分也使用汇编语言),这种用高级语言编写的程序称为“源程序”(Source Program),它最显著的特点是方便人阅读和理解,但是计算机不能直接执行,需要经过一套称为“编译系统”的软件将源程序编译成我们称为目标程序(Object Program)的二进制代码,然后与系统提供的库函数或其他目标代码按照特殊的规则连接起来形成可执行程序。整个过程如图1-1所示。

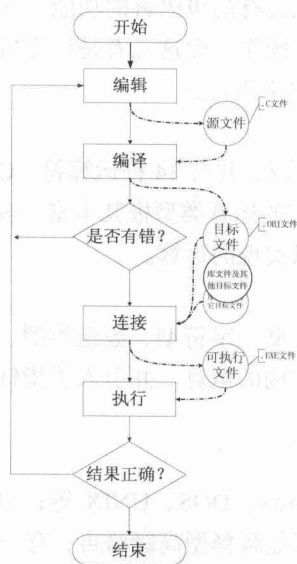


图 1-1 运行 C 程序的流程

1.3.2 C 程序的上机开发过程

开发一个 C 程序需要经过编辑、编译、连接和运行几个步骤，其中编译环节必须要有相应的 C 语言编译系统软件。目前软件厂家都将源程序编辑器、编译软件、连接软件集成到一个软件系统中，这就是我们通常说的集成开发环境（Integrated Development Environment，IDE），这样的软件系统直观易用，使用方便。当然也有没有集成编辑器的纯编译系统，如用于 linux 系统下 C/C++ 编程的编译器 GCC。

可用于 C 语言开发的 IDE 软件系统比较多，如 Microsoft Visual C++、Boland C++ Builder、Borland C++、Watcom C++、GNU DJGPP C++、Lccwin32 C Compiler 3.1、High C、Turbo C、Dev-C++、C-Free 等，不同的开发环境对操作系统及其版本兼容性不同。

学习本课程的目的主要是掌握 C 语言的基本语法规则和程序设计方法，编辑的源程序可以不加任何修改就可以在不同版本的操作系统中利用相关编译系统编译、连接、运行。读者不应该局限于对某一种开发环境的使用，而忽视其他优秀的开发环境。

本节以广泛使用的 Microsoft Visual C++ 6.0 为工具，介绍 C 语言程序的编辑、编译、连接和运行过程。

1. 启动 Visual C++ 集成开发环境

在安装有 Microsoft Visual Studio 软件系统的计算机上，用鼠标顺序选择【开始】→【程序】→【Microsoft Visual Studio】→【Microsoft Visual C++ 6.0】即可。软件运行后，会在屏幕上短暂显示 Microsoft Visual 6.0 的版权信息，然后出现 Microsoft Visual C++ 6.0 主窗口，如图 1-2 所示。

如果桌面上已经建立了 Microsoft Visual C++ 6.0 的快捷方式图标，可以直接双击该图标运行 Microsoft Visual C++ 6.0 集成开发环境。

Microsoft Visual C++ 6.0 主窗口的菜单栏有 9 个菜单项，分别是：

【File】（文件）包括新建文件、打开文件、保存文件等文件相关功能；

【Edit】（编辑）包括拷贝、粘贴、剪切等对文件内容操作的相关功能；

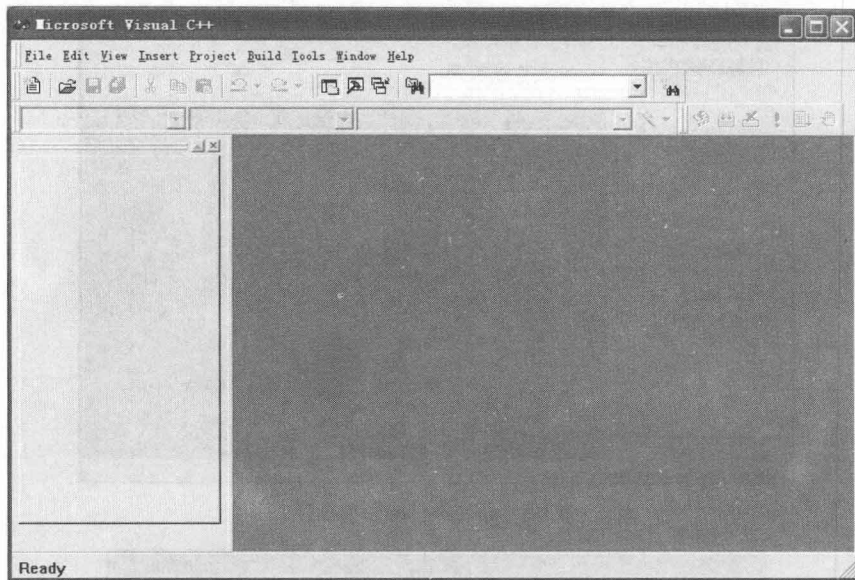


图 1-2 Microsoft Visual C++ 6.0 集成开发环境

【View】(查看)用于配置开发环境中用户常用开发工具的显示方式,如显示工作区、输出窗口等;

【Insert】(插入)包括向正在开发的项目中插入新的类、新的资源(如光标、图标等)等功能;

【Project】(项目)包含创建、修改、保存项目文件在内的多个功能;

【Build】(构建)包括编译、连接、运行应用程序在内的多项功能;

【Tool】(工具)提供用户简单快速地访问各种开发工具,如定制工具栏、菜单项等;

【Window】(窗口)提供对多窗口的快速激活切换和布置方式调整;

【Help】(帮助)向用户提供开发帮助,如用户安装了 MSDN (Microsoft Developer Network),可以为用户提供详细的开发帮助,包括在线帮助。

菜单栏下面是常用工具栏,包括编辑常用工具、调试常用工具等,工具栏显示的工具可以通过【Tool】菜单提供的 Customize 项进行按需要配置。

窗口区左侧是工作区窗口,提供正在开发的工程相关信息;右侧是编辑窗口,显示正在编辑的源文件的内容。

2. 输入/编辑源文件

对于 C 语言的初学者而言,开发一个只包含一个源文件的程序是最简单的。

(1) 新建一个 C 源文件

在 Microsoft Visual C++ 6.0 的菜单栏上单击【File】(文件)菜单,然后在其下拉菜单中选择【New】(新建)项,如图 1-3 所示。

系统立即弹出新建对话框,如图 1-4 所示。在新建对话框中,选择 Files (文件)页,在该页的列表中选择 C++ Source File 项 (Microsoft Visual C++ 6.0 编译器可以编译 C 语言,也可以编译 C++语言);接着在该对话框右侧的 Location (目录)文本框中输入新建的源文件准备保存的路径(如 D:\C Source Files),然后在其上方 File (文件)文本框中输入新建源文件的文件名(如 Hello.c)。文件名的后缀为.c 表示要建一个 C 源程序,如果用户不输入文件名后缀,系统将默认其后缀为.cpp,表示新建的是一个 C++源程序。

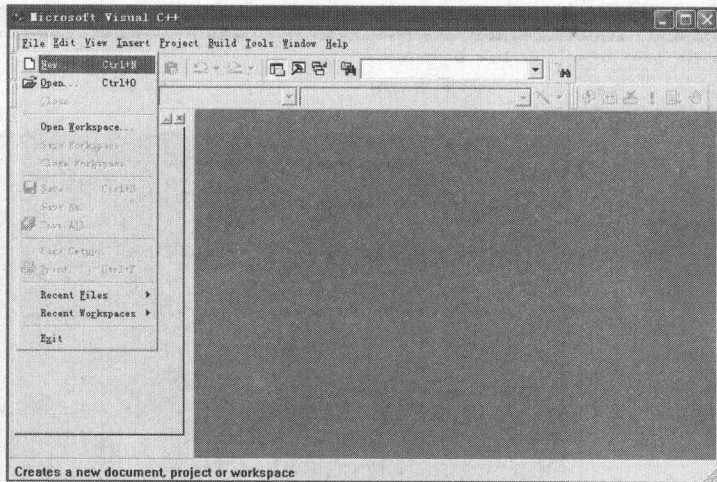


图 1-3 新建文件的菜单选择

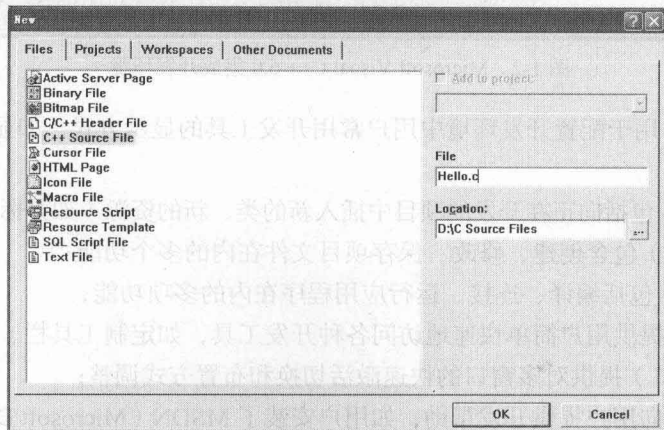


图 1-4 新建 C 源程序的操作

单击 OK (确认) 按钮, 回到主窗口, 编辑窗口中新建的源程序将被激活, 处于编辑状态, 指定文件夹中出现正在编辑的文件。源程序编辑完后单击【File】菜单中的 Save (保存) 项将编辑好的文件存盘, 如图 1-5 所示。当然也可以按快捷键 Ctrl + S 或单击工具栏按钮  对文件进行保存。

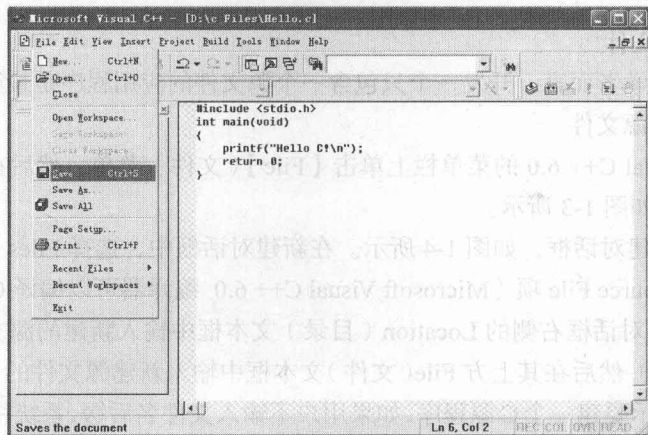


图 1-5 保存文件操作