



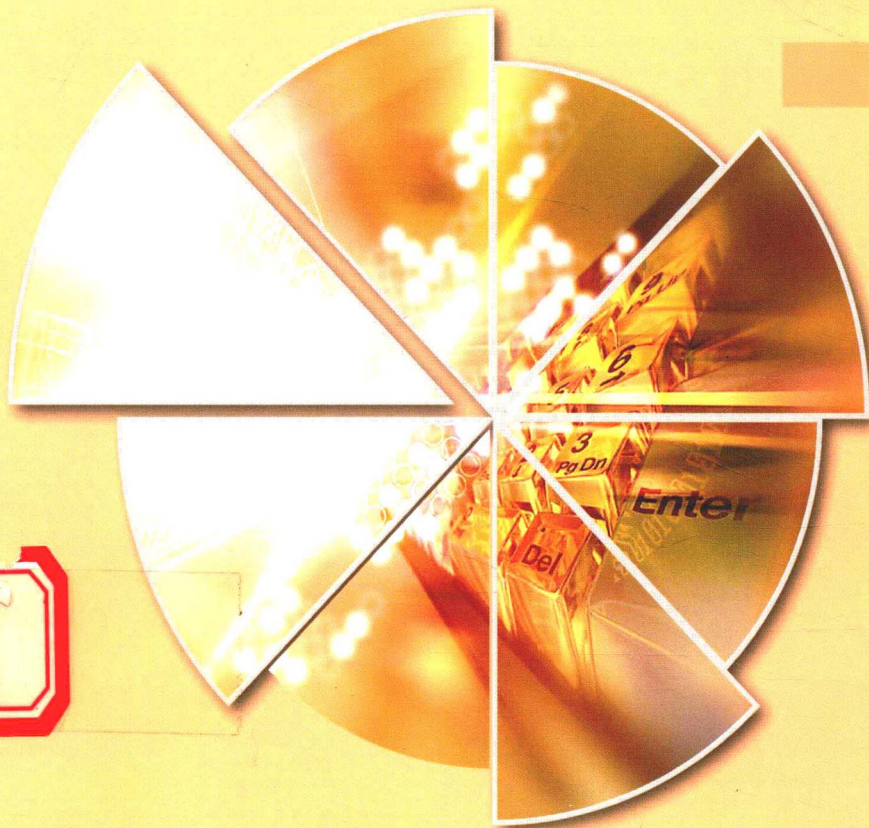
高等教育“十二五”规划教材

C

C YUYAN CHENGXU SHEJI JIAOCHENG

语言程序设计教程

杨文君 蔡春华 刘志宇 主编
赵 杰 主审



科学出版社

高等教育“十二五”规划教材

C 语言程序设计教程

杨文君 蔡春华 刘志宇 主编

赵杰 主审

科学出版社

北京

内 容 简 介

本书是C语言的入门教程,针对初学者没掌握程序设计基础的特点,采用了“提出问题-分析问题-解决问题”的案例教学方式,易于初学者的学习。全书共9章内容,分为三部分,绪论和第1章为C语言程序设计入门部分,介绍了C语言的基础知识及上机运行过程及环境。第2章~第5章是C语言程序设计基础部分,主要介绍了顺序、选择和循环三种程序结构的设计方法。第6章~第9章是C语言程序设计提高部分,主要介绍了有关函数、数组、指针及结构体等模块化结构程序设计和构造类型的程序设计问题。

本书可作为本科院校和高职高专院校各专业学生学习C语言程序设计的入门教程,也可作为C语言程序开发爱好者的自学教材及参考用书。

图书在版编目(CIP)数据

C语言程序设计教程/杨文君等主编. —北京:科学出版社,2010
(高等教育“十二五”规划教材)
ISBN 978-7-03-029800-3

I. ①C… II. ①杨… 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250393号

策划:姜天鹏 李洪旺

责任编辑:王纯刚 隽青龙/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>

百善印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2011年2月第 一 版 开本:787×1092 1/16
2011年2月第一次印刷 印张:12 1/2
印数:1—3 000 字数:293 000

定价:22.00元

(如有印装质量问题,我社负责调换〈百善〉)

销售部电话 010-62140850 编辑部电话 010-62135517-2037

版权所有,侵权必究

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

前 言

C 语言由于其功能强、使用灵活、可移植性好、目标程序质量高等诸多优点，成为国内外应用最广泛的几种计算机语言之一，是程序设计、开发工作者必须掌握的计算机编程语言工具，受到广大计算机用户的喜爱。在我国，几乎所有的理工科大学都开设了 C 语言程序设计方面的课程。该课程具有很强的实践性、设计性和创新性，是大学生综合素质教育和能力培养的重要基础课程。

本书是作者集多年从事 C 语言教学和研究的的工作经验，以“提出问题—分析问题—解决问题”的案例方式进行教学，本着由浅入深的原则，安排各个部分知识点。目的是使学生理解结构化程序设计思路和常用算法，能熟练运用 C 语言实现程序的编制、调试和运行，最终掌握使用 C 语言进行程序设计。

全书共分为三个部分，绪论和第 1 章是 C 语言入门部分，主要介绍 C 程序的基本组成、上机运行过程及环境和 C 语言的基本知识。第 2 章~第 5 章是 C 程序设计基础部分，主要介绍 C 语言程序设计的基本思想和方法及 C 语言程序的三种控制结构。第 6 章~第 9 章是 C 程序设计的提高部分，主要介绍了模块化结构程序设计和构造类型程序设计问题。

本书绪论、第 7 章和第 9 章由杨文君编写，第 1 章~第 4 章由蔡春华编写，第 5 章、第 6 章和第 8 章由刘志宇编写。全书由赵杰主审。

由于水平有限，加之时间仓促，书中错误、疏漏和叙述欠严密之处在所难免。恳请广大读者批评指正。

编 者

2010 年 11 月

目 录

绪论	1
第 1 章 C 语言程序设计的基本知识	4
1.1 C 语言的结构	4
1.1.1 简单的 C 语言程序	4
1.1.2 C 程序的上机运行方法	7
1.2 数据类型	12
1.3 标识符、常量和变量	13
1.3.1 标识符	13
1.3.2 常量和变量	13
1.4 运算符和表达式	21
1.4.1 算术运算符和表达式	21
1.4.2 赋值运算符和表达式	23
1.4.3 关系运算符和表达式	24
1.4.4 逻辑运算符和表达式	26
1.4.5 其他运算符和表达式	27
习题	28
第 2 章 顺序结构	32
2.1 C 语言的语句	32
2.2 数据输入、输出函数	33
2.2.1 格式输出函数 printf ()	33
2.2.2 格式输入函数 scanf ()	37
2.2.3 字符输出函数 putchar ()	40
2.2.4 字符输入函数 getchar ()	40
2.3 顺序结构	41
习题	42
第 3 章 选择结构	47
3.1 用 if 语句实现选择结构	47
3.1.1 if 语句的基本形式	47
3.1.2 if 语句的嵌套	50

3.2	switch 语句	52
	习题	54
第 4 章	循环结构程序设计	60
4.1	while 循环结构	60
4.2	do-while 循环结构	62
4.3	for 循环结构	64
4.4	循环嵌套	68
4.5	break 和 continue 语句在循环体中的作用	70
4.5.1	用 break 语句提前退出循环	70
4.5.2	用 continue 语句提前结束本次循环	71
	习题	71
第 5 章	编译预处理	81
5.1	宏定义	81
5.1.1	无参数宏定义	81
5.1.2	带参数的宏定义	84
5.2	文件包含	88
	习题	89
第 6 章	函数	91
6.1	函数的定义	92
6.2	函数调用	94
6.3	函数的返回值	99
6.4	函数的说明	99
	习题	100
第 7 章	数组	107
7.1	一维数组	107
7.1.1	一维数组的定义	107
7.1.2	一维数组元素的引用	108
7.1.3	一维数组的初始化	109
7.1.4	一维数组的应用举例	110
7.2	二维数组	114
7.2.1	二维数组的定义	114
7.2.2	二维数组元素的引用	115

7.2.3 二维数组的初始化	116
7.2.4 二维数组的应用举例	117
7.3 字符串与字符数组	118
7.3.1 字符串	118
7.3.2 字符数组	119
7.3.3 用于字符处理的函数	122
7.3.4 字符串处理程序设计举例	124
7.3.5 二维字符数组	126
习题	127
第8章 指针	131
8.1 变量的地址与指针	131
8.2 指针变量的定义和指针变量的基本类型	133
8.3 指针变量的赋值	133
8.4 指针变量的操作	136
8.5 指针变量作为函数参数	139
8.6 指向一维数组的指针变量	141
8.6.1 一维数组指针的概念	141
8.6.2 数组元素地址作实参	142
8.6.3 数组名作函数参数	143
8.6.4 函数的指针形参和函数体中数组的区别	145
8.7 指向二维数组的指针变量	145
8.7.1 二维数组地址的概念	145
8.7.2 通过建立一个行指针来引用二维数组元素	147
8.7.3 二维数组名作为实参时, 实参和形参之间的数据传递	147
8.8 字符串与字符串指针变量	150
8.9 指针数组	152
8.9.1 指针数组的定义及引用	152
8.9.2 指针数组作为实参时, 实参和形参之间的数据传递	154
8.10 指向指针的指针	154
8.11 指针数组作 main 函数的形参	156
8.12 指针与函数	157
8.12.1 指向函数的指针变量	157
8.12.2 返回指针值的函数	159
习题	159



第9章 结构体及共用体	163
9.1 用 typedef 定义一种新类型	163
9.2 结构体类型	164
9.2.1 结构体类型的定义	165
9.2.2 结构类型变量的说明	166
9.2.3 对结构体变量的赋值、初始化及引用	168
9.2.4 结构体数组的定义、初始化及引用	171
9.2.5 结构指针变量的说明和使用	172
9.2.6 结构体变量作为函数参数及返回结构体类型值的函数	176
9.2.7 用指针处理链表	178
9.3 共用体	183
9.3.1 共用体类型的定义和变量的说明	183
9.3.2 共用体变量的引用	184
习题	185
主要参考文献	189

绪 论

C 语言是目前国际上广泛使用的通用计算机高级语言。既可用它编写计算机系统软件，也可编写各种应用软件，因此在数百种计算机语言中，C 语言是学习和使用人数最多，最受欢迎的一种计算机语言。

最初的 C 语言只是为编写 UNIX 操作系统而提供的一种工作语言，其产生颇为曲折。原型是 ALGOL 60 语言，但 ALGOL 60 是面向问题的高级语言，不宜用来编写系统程序。

1963 年，英国剑桥大学在 ALGOL 60 程序设计语言基础上，推出了更接近于硬件的 CPL (Combined Programming Language) 语言。由于 CPL 规模大、实现难等原因，1967 年 Martin Richards 对 CPL 作了简化，推出了 BCPL (Basic Combined Programming Language) 语言。如图 0-1 所示为 BCPL 语言的设计者 Martin Richards。

1970 年，美国贝尔实验室的 Ken Thompson 又在 BCPL 的基础上，再次作进一步的简化，设计出既简单又接近硬件的 B 语言，并用 B 语言写了 UNIX 操作系统和大量的实用程序。如图 0-2 所示为 B 语言的设计者 Ken Thompson。

由于 B 语言只有单一的字类型，过于简单等原因而未能广泛使用。D. M. Ritchie 从 1971 年开始在 B 语言基础上设计了 C 语言，并于 1972 年投入使用。如图 0-3 所示为 C 语言的设计者 D. M. Ritchie。

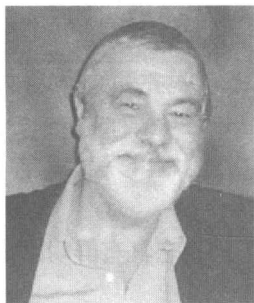


图 0-1 BCPL 语言的设计者
Martin Richards

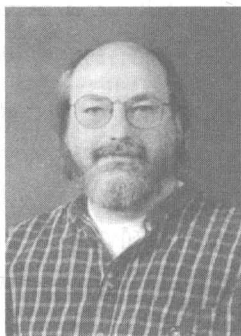


图 0-2 B 语言的设计者
Ken Thompson

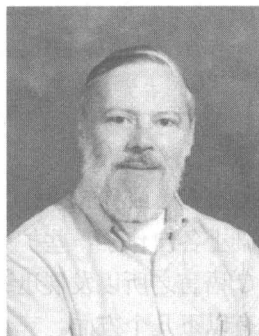


图 0-3 C 语言的设计者
D. M. Ritchie

1973 年，K. Thompson 和 D. M. Ritchie 两人合作把 UNIX 系统用 C 语言重写了一遍，而原来的 UNIX 操作系统则是 K. Thompson 和 D. M. Ritchie 于 1969 年使用汇编语言开发成功的。这次用 C 语言编写的第五版 UNIX 操作系统增加了多程序设计功能，使整个系统（包括 C 语言的编译程序）都建立在 C 语言的基础上，在移植性和代码效率方面达到了很好

的平衡。

后来C语言多次进行了改进,但主要还是在贝尔实验室内部使用。直到1975年,UNIX第六版问世。随着UNIX的巨大成功和广泛应用,C语言也逐渐被人们所接受,并迅速得到推广。1977年,出现了不依赖于具体机器的C语言编译文本“可移植C语言编译程序”,使C语言移植到其他机器时所做的“后勤工作”大大简化,这也促使了UNIX操作系统迅速地在各种计算机上应用。因此,可以说C语言和UNIX操作系统是一对孪生兄弟,在发展过程中相辅相成。C语言的具体发展历程如图0-4所示。

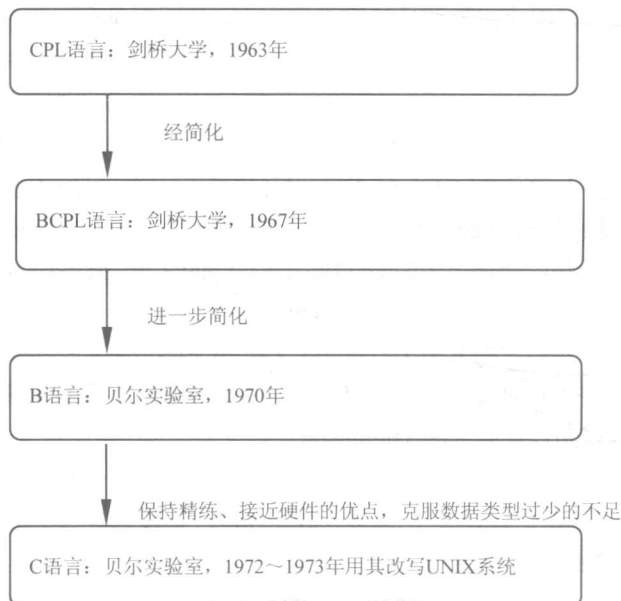


图0-4 C语言的发展历程

1983年,美国国家标准化协会(ANSI)对C语言的各种版本作了扩充和完善,制定了C的标准,称为ANSI C。目前最流行的C语言版本有Borland Turbo C(或称Turbo C)、Microsoft C(或称MS C)和AT&T C。这些C语言版本不仅实现了ANSI C标准,而且在此基础上各自作了一些扩充,使之更加方便和完善。

C语言之所以发展迅速,成为风靡全球的程序设计语言,主要是因为它具有鲜明的特点,具体有以下几个方面。

(1) 语言表达能力强。

C语言包含34种运算符,有的运算符反映了当前计算机的性能,包含可直接由硬件实现的算术逻辑运算,有效到足以取代汇编语言编写各种系统程序和应用程序。众多的运算符使C的运算类型极其丰富,可以表达数值运算、字运算、位运算和地址运算等。

(2) 含有丰富的数据类型,具有类型构造能力和结构化的程序控制结构。

C语言能在字符、整数、浮点数等基本类型基础上按结构化的层次构造方法构造数组、结构体和共同体等各种数据类型。特别是C语言的指针类型灵活多样,非常有助于构造链

表、树、图等复杂的数据结构。另外，它的结构化程序控制结构符合结构化程序设计的要求，可编写出结构非常良好的程序。此外，它的数据静态和外部存储类机制有助于信息隐蔽和抽象的模块化程序设计。

(3) 语言简洁、紧凑、使用方便灵活。

用 C 语言编写的程序通常比用其他高级语言编写的程序更简练，代码行少。C 语言没有 I/O 设施，也没有并行操作、同步或协同程序等复杂控制。另外，C 语言程序在运行时所需要的支持少，占用的存储空间也小。

(4) 能使编译程序产生执行效率高的代码。

一个高级语言能否用来描述系统程序，除语言表达能力之外，还有一个重要因素就是能否产生高质量的代码。许多高级语言相对汇编语言而言其代码的执行效率要低得多。但 C 语言则不然，大量的试验表明，用 C 语言描述较汇编语言描述，其代码执行效率只低 10%~20%，而用 C 语言编程比汇编语言编程速度要快得多，且程序的可读性高，特别是 C 语言程序比较容易移植。所以 C 语言成了人们描述系统程序和应用程序比较理想的工具。

(5) 用 C 语言编写的程序可移植性较好。

程序的可移植性是指在一个环境上运行的程序可以不加或稍加改动后即可在另一个完全不同的环境上运行。汇编语言是依赖于机器硬件的，用汇编语言编写的程序不可移植。同样，有些高级语言，由于它们的编译程序不可移植，影响了用它们编写的程序的可移植性。而目前在许多机器上都有 C 编译系统，且大部分是由 C 语言编译移植得到的。由于 C 语言的编译程序便于移植，也就大大提高了 C 程序的可移植性。

(6) C 语言有诸多优点，但也有一些不足之处。

例如，用 C 语言编写程序，自由度大（如对变量的类型约束不够严格）、整型和字符型及逻辑型数据的通用、指针和数组的通用等。过多的通用性限制了编译程序对 C 程序作充分的句法和语义检查，可能会无视某些使用上的失误，依旧能正常编译，从而不能及时发现程序中的错误，给程序的调试和排错造成一些困难。另外，C 语言的运算符优先级太多，不便于记忆，有些还与常规约定有所不同；类型检验弱、数据类型转换比较随意等也影响了程序的安全性。

第 1 章 C 语言程序设计的基本知识

1.1 C 语言的结构

1.1.1 简单的 C 语言程序

下面我们先介绍几个简单的 C 语言程序，然后从中分析 C 语言程序的基本结构以及所涵盖的基础语法知识，从而对 C 语言程序产生初步认识。

例 1.1: 要求在屏幕上输出以下信息。

```
Welcome to C world!
```

程序如下:

```
#include <stdio.h>                /*预处理命令*/
void main( )                       /*主函数*/
{
    printf("Welcome to C world! \n"); /*格式输出函数*/
}
```

程序说明:

(1) 该程序是一个最简单的 C 语言程序，在程序的第二行中，main 是 C 语言程序中的“主函数”的名字。main 前面的 void 表示此函数是“空类型”，即执行此函数后不产生函数值。在 C 语言中，每个程序都必须有一个 main 函数，它是每一个 C 语言程序的执行起始点（入口点），也是程序执行的结束点。

(2) 用 {} 括起来的是 main 的函数体。main 函数中的所有操作（或语句）都在这一对 {} 之间。也就是说，main 函数的所有操作都在 main 函数体中。

(3) 本例主函数中只有一条语句，printf 是 C 语言库函数中的格式输出函数，功能是把该语句中双撇号内的字符原样输出（显示在屏幕上），“\n”是换行符，即在输出“Welcome to C world!”后回车换行。语句最后必须有一个分号。

(4) 在 C 语言中，使用标准函数库中的输入输出函数时，编译系统要求程序提供有关信息（即对这些输入输出函数的声明），程序第 1 行“#include <stdio.h>”的作用就是用来提供这些信息的，“stdio.h”是 C 编译系统提供的一个标准输入输出的库文件。include 是文件包含命令。对这一行我们可暂不必深究，以后会在第 5 章中详细介绍，在此只须记住：在程序中，用到系统提供的标准函数库中的输入输出函数时（即本例中的 printf 输出函数），就需要在程序的开头写上这样一行：

```
#include <stdio.h>
```

(5) “/* */”表示注释部分，注释可以用汉字或英文字符表示，可以出现在语句的后面，也可以单独成为一行。在程序编译和运行过程中不起作用，只是为了便于别人读懂程序。

例 1.2: 求两数中的最小值。

程序如下:

```
#include <stdio.h>
void main( ) /* 主函数 */
{
    int a , b, c; /* 定义变量 */
    scanf("%d, %d", &a, &b); /* 输入数 */
    c=min(a, b); /* 调用函数 */
    printf("min=%d\n", c); /* 输出 */
}
int min(int x, int y) /* 定义函数 */
{
    int z; /* 定义局部变量 */
    if (x>y)
        z=y;
    else
        z=x;
    return(z); /* 返回 z 值*/
}
```

经编译后运行如下:

```
14, 4 ✓
min=4
```

再次运行:

```
6,29 ✓
min=6
```

程序说明:

(1) 本程序包括两个函数: 主函数 main() 和子函数 min()。min 函数的作用是将 x、y 中较小的数送给变量 z, return 语句的作用是将 z 的值返回给主函数 main。返回值是通过函数名 min 带回给 main 中的调用处。

(2) 主函数中的 scanf() 是系统给出的标准输入函数, 通过它输入 a 和 b 的值, &a、&b 中的“&”是取地址符, 在此功能是将输入的值分别送入变量 a 和 b 的地址所标志的存储单元中, 即输入给变量 a 和 b。“%d, %d”是输入输出的格式字符串, 用来指定输入输出数据的类型和格式。

(3) main 函数中第 4 行是调用 min 函数, 在调用时将实际参数 a、b 的值送给 min 函数中的形式参数 x 和 y。在 min 函数后得到一个返回值 (即 min 函数中的变量 z 的值), 把这个值送给变量 c, 然后输出变量 c 的值。

通过上述两个例子，我们可以看到：

(1) C 程序是由函数构成的。一个程序至少含有一个 `main` 主函数。也可以包含一个主函数和若干个自定义函数。函数是 C 语言的基本单位。被调用的函数可以是系统给出的库函数，也可以是用户自己定义的函数。

(2) 一个函数由两部分组成：函数的首部和函数体。

① 函数首部。即函数的第一行。包含函数类型、函数名、函数属性、函数参数名（形式参数）、参数类型等。

以例 1.2 中的 `min` 函数来介绍，可以看到如图 1-1 所示的对应关系。

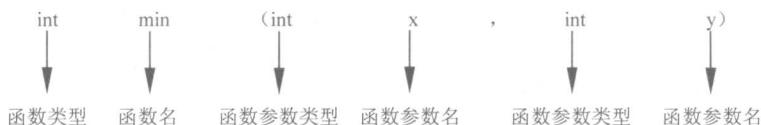


图 1-1 函数结构介绍

函数可以没有参数，但是函数后必须带一对圆括弧，如：

```
void main()
```

② 函数体。即函数首部下面花括弧中的内容，当一个函数内有多个大括号，则最外层的一对大括号为函数体的范围。

函数体一般包含声明和执行两部分。

- 声明部分用于定义函数中所使用的变量和对所调用函数的声明，如例 1.2 中 `main` 函数体中的 `int a, b, c;`
- 执行部分由若干条语句组成。

在某些情况下也可以没有声明部分，甚至也没有执行部分。例如：

```
void main()
{ }
```

也是一个合法的 C 语言程序。

(3) `main` 函数通常位于程序之首，实际上它位于程序的开头、最后及函数与函数之间均是合法的，但不管在什么位置，一个 C 程序总是从 `main` 函数开始执行的。

(4) C 程序书写格式自由，一行内可以写几条语句，一条语句可以分写在多行上。例如：

```
int a,b;
a=5; b=8; /*两条语句在一行*/
printf("a=%d,b=%d\n",
    a,b); /*一条语句在两行*/
```

(5) C 语言本身没有输入输出语句，输入输出都由函数实现，因此在程序的开始必须有头文件“`include <stdio.h>`”。

(6) 可以用 `/* ... */` 对 C 程序中的任何部分作注释，程序中加入必要的注释可增加程序的可读性。

(7) C语言程序中每一条语句结束后都必须加一个分号。分号是C程序语句的组成部分。

1.1.2 C程序的上机运行方法

在编写好一个C源程序后，如何上机运行呢？这就要有相应的C语言编译系统。

IBM PC 微机 DOS、Windows 平台上常见的C语言版本有：

- Borland 公司：Turbo C、Turbo C++、Borland C++、C++ Builder（Windows 版本）
- Microsoft 公司：Microsoft C、Visual C++（Windows 版本）

我们以 Visual C++ 6.0 为例，介绍 C 程序在该环境下如何进行编辑、编译、连接和运行。

1. Visual C++ 的安装与启动

本节介绍的是 Visual C++ 6.0 中文版。Visual C++ 6.0 是在 Windows 环境中工作的。

为了能使用 Visual C++ 6.0 集成环境，必须事先在所用的计算机上安装 Visual C++ 6.0 系统。安装后最好在桌面上建立 Visual C++ 6.0 的快捷方式图标，以方便使用。

双击桌面上 Visual C++ 6.0 图标，就能进入 Visual C++ 6.0 集成开发环境，Visual C++ 6.0 的主窗口如图 1-2 所示。

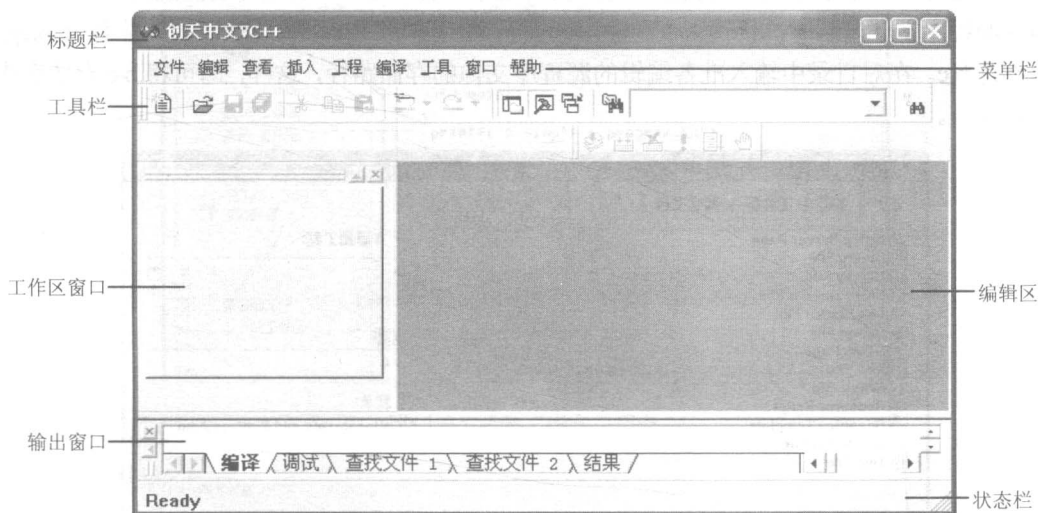


图 1-2 Visual C++ 6.0 主窗口

2. 输入与编辑源程序

(1) 新建一个源程序。

在 Visual C++ 6.0 主菜单栏中选择“文件”|“新建”命令，如图 1-3 所示。

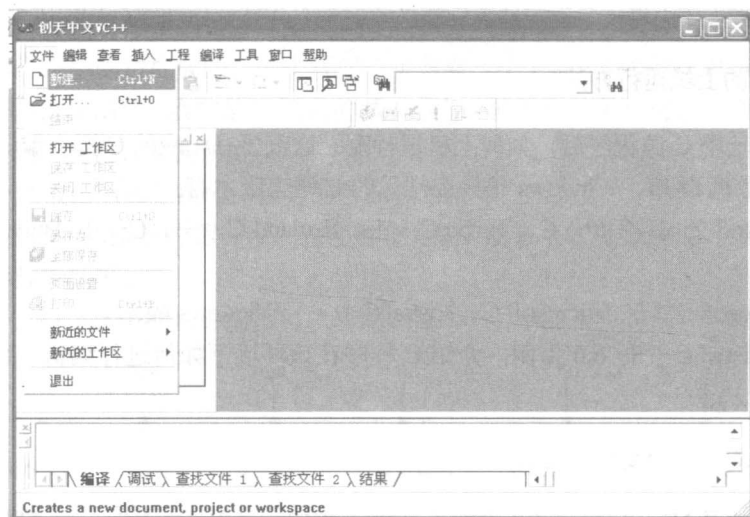


图 1-3 选择“新建”命令

此时弹出如图 1-4 所示的“新建”对话框。选择对话框中的“文件”选项卡，在其列表框中选择 C++ Source File 选项，表示要建立一个新的 C++ 源程序，然后在对话框右半部分的“文件”文本框中输入编辑的源程序文件名字（如 b.c），这里需要注意的是：如果在输入源程序文件名字时，仅输入文件主名 b 而省略了扩展名.c，系统默认的文件名为 b.cpp 而不是 b.c。在对目录中输入准备编辑的源程序文件的存储路径，这样，即可将 b.c 存放在指定目录中。

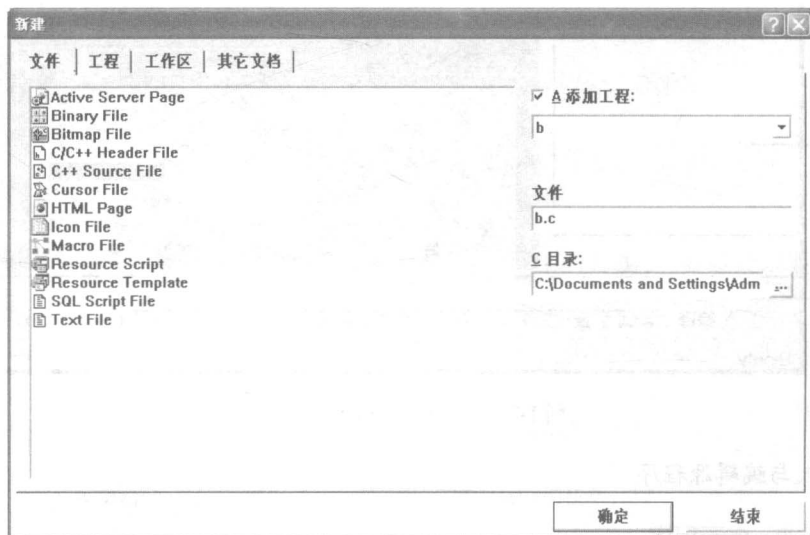


图 1-4 “新建”对话框

单击“确定”按钮后，即可在编辑区输入源程序，如图 1-5 所示。



图 1-5 在编辑区输入源程序

注意：源程序一定要在英文状态下输入，即字符标点都要在半角状态下，同时注意大小写，一般都用小写。若输入过程中出现错误，可以利用全屏编辑方法进行个性编辑。

输入完毕后，在主菜单栏中选择“文件”|“保存”命令，保存文件，如图 1-6 所示。



图 1-6 选择“保存”命令

(2) 程序的编译。

在输入和保存了源文件以后，选择主菜单栏中的“编译”|“编译 b.c”项，如图 1-7 所示。

此时屏幕上会出现一个提示对话框，此编译命令要求一个有效的项目工作区，问用户是否同意建立一个默认的项目工作区，如图 1-8 所示，单击“是”按钮，表示同意由系统建立默认的项目工作区，即可开始编译。也可以不用通过选择菜单的方法，而使用快捷键 Ctrl+F7 来完成编译。编译系统会检查源程序中是否有语法错误，然后在主窗口下部的调试信息窗口输出编译的信息，如果没有错误，则生成目标文件 b.obj。否则会指出错误的位置和性质，提示用户改正错误。