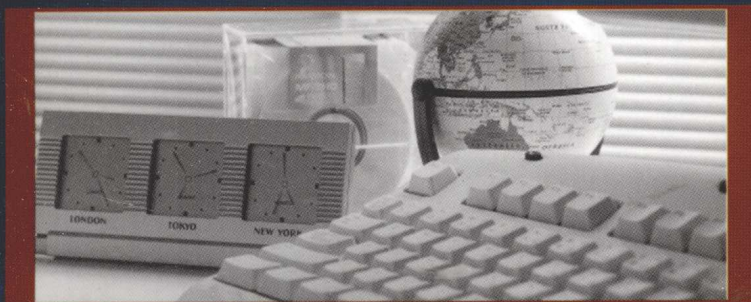


高职高专教育“十二五”规划教材

C 语言程序设计

C YUYAN CHENGXU SHEJI

主编 郭丽春



北京工业大学出版社

高职高专教育“十二五”规划教材

C 语言程序设计

主 编：郭丽春

副主编：郑淑荣 赵丽娟 李忠濡

张 森 成安霞

参 编：罗雨滋 李灵佳 贾敬典

北京工业大学出版社

内 容 简 介

本书是依据高等职业技术教育人才培养的基本要求编写的一本实践性很强的教材。全书共分9章,内容包括:C语言概述,数据类型、运算符及表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数,预处理,指针,结构体与共用体,文件,应用程序的开发等。每章均有精心设计的习题和上机实训指导,习题内容知识点全面、题量适中、题型多。实训部分的主要目的是培养学生的程序设计能力。

本书可作为高职高专计算机专业和非计算机专业开设的C语言程序设计课程教材,也可作为其他人员学习C语言程序设计的参考书。

图书在版编目(CIP)数据

C语言程序设计/郭丽春主编. —北京:北京工业大学出版社, 2010.8

ISBN 978-7-5639-2439-4

I. ①C… II. ①郭… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第150520号

C语言程序设计

主 编: 郭丽春

责任编辑: 王铁杰

出版发行: 北京工业大学出版社

地 址: 北京市朝阳区平乐园100号

邮政编码: 100124

电 话: 010-67391106 010-67392308 (传真)

电子信箱: bgdcbsfxb@163.net

承印单位: 北京溢漾印刷有限公司

经销单位: 全国各地新华书店

开 本: 787 mm×1 092 mm 1/16

印 张: 15.75

字 数: 390千字

版 次: 2010年8月第1版

印 次: 2010年8月第1次印刷

标准书号: ISBN 978-7-5639-2439-4

定 价: 32.00元

版权所有 翻印必究

图书如有印装错误, 请寄回本社调换

前 言

高职高专类学校人才培养的主要目标是培养技能型和应用型的人才。《C 语言程序设计》是依据高等职业技术教育人才培养的基本要求而编写的一本实践性很强的教材。全书以实用为主,以知识够用为原则,力求学以致用。每章中的实训是基本理论知识的延伸和拓展,其目的是强化训练,培养学生的逻辑思维能力和通过编程解决实际问题的能力,从而实现知识能力向职业行为能力的转化。

程序设计是非常重要的专业基础课,同时又是一门理论性和逻辑性强、比较抽象的课程,为了引导学生学好这门课程,根据多年的教学经验,在教材的编写上着重注意以下一些方面:

(1) 以 Visual C++6.0 为开发环境。源代码均在 Visual C++6.0 下运行通过,所有实例的输出结果均通过屏幕拷贝后截取而获得,可以确保源代码的正确性。

(2) 将编程的理论和方法融入案例中。结合高职高专学生的特点,将 C 语言的基本概念、基本理论和编程的基本方法都尽量放在程序案例中,各个程序案例不仅有详细的分析和注释,而且有完整的输入和输出结果显示。

(3) 以项目带动案例,强调 C 语言程序设计的实用性。在本书第 9 章 C 语言课程设计中,通过完整的“学生成绩管理系统”项目的分析研究、综合设计,将前面各章的内容组织起来,提高了学生 C 语言程序设计的综合运用能力。

(4) 根据高职高专计算机教育的目标,组织教学内容和知识结构,以应用为主线,力求做到科学、全面、先进、合理。教学内容及结构体现了知识的渐进性、系统性、逻辑性和科学性的规律。各章前导部分有教学提示、教学要求,各章的结束部分有本章小结、习题和实训。

(5) 易于教师的讲授和学生的自主学习。

本书参编人员均是一线的专职教师,有多年讲授本课程的实际经验,因此,可以确保教材内容的正确性和实用性。

本书课程的建议总体学时为 64~72 学时,同时建议增加一周的课程设计时间,各学校可以根据本校的教学大纲对讲授内容进行适当选择,理论讲授学时和上机实验学时比例为 1:1。

本书主要作为高等职业技术学院学生“C 语言程序设计”课程的教学用书,也可以作为参加全国计算机等级考试及各类短训班的人员的辅导教材或自学参考书。

由于时间仓促,加之水平有限,书中难免出现疏漏甚至错误之处,恳请读者批评指正。

目 录

第 1 章 C 语言概述	1
1.1 程序设计语言	1
1.2 简单的 C 程序介绍	2
1.3 C 语言的基本词法	5
1.4 C 程序的运行环境	8
本 章 小 结	18
习 题	19
实 训	20
第 2 章 数据类型及表达式	22
2.1 C 语言的基本数据类型	22
2.2 常量和变量	24
2.3 运算符和表达式	30
2.4 实例解析	40
本 章 小 结	42
习 题	42
实 训	44
第 3 章 程序设计结构	46
3.1 顺序结构程序设计	46
3.2 选择结构程序设计	54
3.3 循环结构程序设计	62
3.4 实例解析	74
本 章 小 结	77
习 题	77
实 训	79
第 4 章 数 组	83
4.1 一维数组	83
4.2 二维数组	85
4.3 字符数组和字符串	87
4.4 实例解析	92
本 章 小 结	100
习 题	100
实 训	102



第 5 章 函数与预处理	105
5.1 函数概述	105
5.2 函数的定义	107
5.3 函数的参数和返回值	111
5.4 函数的调用	114
5.5 变量的作用域与存储类别	120
5.6 内部函数与外部函数	128
5.7 编译预处理	130
5.8 实例解析	136
本章小结	140
习 题	141
实 训	142
第 6 章 指 针	146
6.1 指针的概念	146
6.2 指针和指针变量	147
6.3 指针与数组	154
6.4 指针与字符串	164
6.5 指针数组	166
6.6 指向函数的指针	168
6.7 实例解析	172
本章小结	175
习 题	176
实 训	178
第 7 章 构造数据类型	182
7.1 结构体	182
7.2 指向结构体类型数据的指针	187
7.3 共用体	194
7.4 枚举类型	197
7.5 用 typedef 定义类型	200
本章小结	201
习 题	202
实 训	205
第 8 章 文 件	208
8.1 文件的概述	208
8.2 文件的打开与关闭	209
8.3 文件的读写	211
8.4 文件的定位	213
8.5 文件的出错检测	214
8.6 实例解析	215



本章小结	222
习 题	222
实 训	223
第 9 章 C 语言课程设计	225
9.1 应用程序设计方法	225
9.2 学生成绩管理系统设计实例	226
本章小结	235
习 题	235
实 训	236
附 录	237
参 考 文 献	244

第1章 C语言概述

教学提示

C语言是国际上广泛流行的计算机高级程序设计语言,它集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也可用于应用软件的开发,同时它还具有效率高、可移植性等特点。本章主要介绍C语言的结构特点、程序组成、书写规则、上机运行过程和调试应用程序的方法。

教学要求

通过本章的学习,要求学生能理解C语言程序的构成、C语言的词法规定和书写规范,掌握C程序的上机步骤和C程序的运行环境。

1.1 程序设计语言

人与计算机交流信息使用的语言称为计算机语言或程序设计语言,通常分为机器语言、汇编语言和高级语言。

最初程序员使用的程序设计语言是一种用二进制代码“0”和“1”形式表示的,能被计算机直接识别和执行的语言。该语言称为机器语言,是一种低级语言,用机器语言编写的程序不便于记忆、阅读和书写。所以,现在通常不用机器语言直接编写程序。

在机器语言的基础上,设计出了汇编语言可以将机器语言用便于人们记忆和阅读的助记符表示,如ADD、SUB、MOV等。计算机运行汇编语言程序时,将用助记符写成的源程序转换成机器语言程序后才能运行。汇编语言适用于编写直接控制机器操作的低层程序,它与机器密切相关。汇编语言和机器语言都是面向机器的程序设计语言,因而称为低级语言。

随着计算机应用的发展,出现了高级程序设计语言。高级程序设计语言是一种与硬件结构及指令系统无关,表达方式比较接近日常语言和数学表达式的计算机程序设计语言。用高级语言编写的程序称为“源程序”,计算机不能识别和执行,而需要把用高级语言编写的源程序翻译成机器指令,通常有编译和解释两种方式。

编译方式是将源程序整个编译成等价的、独立的目标程序,然后通过链接程序将目标程序链接成可执行程序。解释方式是将源程序逐句翻译,翻译一句执行一句,边翻译边执行,不产生目标程序,在整个执行过程中,解释程序都一直在内存中。

C语言是一种具有很高灵活性的高级程序设计语言。1972年至1973年间,贝尔实验室

的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言,后来对 C 语言又做了多次改进。早期的 C 语言主要是用于 UNIX 系统。由于强大功能和各方面的优点,C 语言逐渐为人们所认识。到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用,成为当代最优秀的程序设计语言。

C 语言是一种结构化语言。它层次清晰,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,不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,它还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于实现了对硬件的编程操作,因此 C 语言集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也适合于应用软件的开发。此外,C 语言还具有效率高、可移植性强等特点。

1.2 简单的 C 程序介绍

1.2.1 三个简单的 C 程序实例

用 C 语言语句编写的程序称为 C 程序或 C 源程序。下面先介绍三个简单的 C 程序,这几个程序都是在 Visual C++ 6.0 环境下编译通过的。

【例 1.1】 用 C 语言编写一个程序,输出“C 语言世界,您好!”。

程序代码如下:

```
#include <stdio.h>

void main( )                /* 定义主函数 */
{
    printf("C 语言世界,您好! \n");    /* 输出 C 语言世界,您好! */
}
```

说明:

(1) #include <stdio.h> 是一条编译预处理命令,声明该程序要使用 stdio.h 文件中的内容,stdio.h 文件中包含了输入函数 scanf() 和输出函数 printf() 的定义。编译时系统将头文件 stdio.h 中的内容嵌入到程序中该命令位置。C 语言中编译预处理命令都以“#”开头。C 语言提供了 3 类编译预处理命令:宏定义命令、文件包含命令和条件编译命令。例 1.1 中出现的 #include <stdio.h> 是文件包含命令,其中尖括号内是被包含的文件名。

(2) 程序中定义了一个主函数 main(), 其中 main 是函数名,void 表示该函数的返回值类型。程序执行从主函数开始。一个 C 语言的程序可以包含多个文件,每个文件又可以包含多个函数。函数之间是相互平行、相互独立的。一个 C 程序,必须有一个且只能有一个主函数 main()。执行程序时,系统先从主函数开始运行,其他函数只能被主函数调用或通过主函数调用的函数进行调用,函数还可以嵌套调用,即在一个函数中调用另外一个函数。主函数可以带参数,也可以不带参数。函数在调用之前,必须先定义好,定义函数要按照系统规定的格式进行,后面再详细介绍。

(3) 用 {} 括起来的是主函数 main() 的函数体。main 函数中的所有操作语句都在这一



对{}之间。即 main 函数中的所有操作都在 main 的函数体中。

(4) printf 函数是一个由系统定义的标准函数,可在程序中直接调用,printf 函数的功能是把要输出的内容送到显示器去显示,双引号中的内容要原样输出。“\n”是换行符,即在输出完“C 语言世界,您好!”后回车换行。

(5) 每条语句用“;”表示结束。

(6) /* */括起来的一部分是一段注释,注释只是为了改善程序的可读性,是对程序所需部分的说明,向用户提示或解释程序的意义。/* 是注释的开始符号,* /是注释的结束符号,这两个符号必须成对使用。程序编译时,计算机不对注释作任何处理。注释可出现在程序中的任何位置。

程序运行后,屏幕显示:

c语言世界,您好!

【例 1.2】 计算两数之和,并输出结果。

程序代码如下:

```
#include <stdio.h>

void main()
{
    int a,b,sum; /* 定义三个整型变量 */
    a=123; b=456; /* 给变量 a,b 分别赋值为 123,456 */
    sum=a+b; /* 变量 a 的值加上变量 b 的值,然后将两数的和赋给变量 sum */
    printf("sum is %d\n",sum); /* 输出变量 sum 的值 */
}
```

说明:

(1) 此程序同样也包含了一条 #include <stdio.h> 编译预处理命令,以一个 main 函数作为程序执行的起点。

(2) “int a,b,sum;”是变量声明,声明了三个具有整数类型的变量 a,b,sum。因为 C 语言的变量必须先声明再使用。

(3) “a=123; b=456;”是两条赋值语句。将整数 123 赋给整型变量 a,将整数 465 赋给整型变量 b。每条语句均用“;”结束。

也可将两条语句写成两行,即

```
a=123;
b=456;
```

由此可见 C 语言程序的书写可以相当随意,但是为了保证容易阅读要遵循一定的规范。

(4) “sum=a+b;”是将 a,b 两变量相加,然后将结果赋值给整型变量 sum。

(5) “printf(“sum is \n”,sum);”是调用库函数 printf() 输出 sum 的结果。“%d”为格式控制,表示 sum 的值以十进制整数的形式输出。

程序运行后,屏幕显示:

sum is 579

【例 1.3】 编写程序,要求先输入两个整数,再输出其中较大的数。

程序代码如下:

```
#include <stdio.h>

void main()                /* 主函数 */
{
    int max(int x,int y);    /* 对被调函数 max 的声明 */
    int a,b,c;              /* 定义变量 a、b、c */
    printf("请输入 a 和 b 的值:"); /* 提示输入 a 和 b 的值 */
    scanf("%d%d",&a,&b);      /* 从键盘输入变量 a 和 b 的值 */
    c=max(a,b);             /* 调用 max 函数,将得到的值赋给 c */
    printf("max= %d\n",c);    /* 输出 c 的值 */
}

int max(int x,int y) /* 定义 max 函数,函数值为整型,形式参数 x、y 为整型 */
{
    int z; /* max 函数中的声明部分,定义本函数中用到的变量 z 为整型 */
    if(x>y) z=x;
    else z=y;
    return (z); /* 将 z 的值返回,通过 max 带回调用处 */
}
```

说明:

(1) 本程序包括两个函数。一个是程序的主函数 main,另一个是被调用的函数 max。max 的作用是将 x 和 y 中较大者的值赋给变量 z。return 语句将 z 的值返回给主调函数 main。程序第 4 行 是对被调函数的 max 的声明,max 函数是一个用户自定义函数,由于在 main 函数中要调用 max 函数,而 max 函数的位置在 main 函数之后,必须在调用 max 函数之前对 max 函数进行声明。

(2) scanf 是“输入函数”的名字,它的作用是从键盘上输入 a、b 的值。&a 和 &b 中的“&”的含义是“取地址”。此 scanf 函数的作用是:将键盘上输入的两个数值分别输入到变量 a、b 的地址所标志的单元中,也就是输入给变量 a、b。

例如,输入 a、b 的值分别为 45、89,程序运行后,屏幕显示:

```
请输入a和b的值: 45 89
max= 89
```

1.2.2 C 语言程序的构成和书写规则

1. C 语言程序的构成

(1) C 程序是由函数构成的,函数是 C 程序的基本单位。一个源程序至少包含一个 main 函数,即主函数,也可以包含一个 main 函数和若干个其他函数。被调用的函数可以是系统提供的库函数,也可以是用户根据需要自己编写设计的函数。

(2) main 函数是每个程序执行的起始点。C 程序总是从 main 函数开始执行,不管 main 函数在程序中的任何位置。所以,可以将 main 函数放在整个程序的最前面,也可以放在整个程序的最后,或者放在其他函数之间。

(3) 源程序可以有预处理命令,预处理命令通常放在源文件或源程序的最前面。

(4) 每个语句都必须以分号结尾,但预处理命令、函数头和花括号“{}”之后不加分号。



(5) 标识符和关键字之间,至少加一个空格以示间隔,空格的数目不限。

(6) 源程序中需要解释和说明的部分,可用“/* */”加以注释,注释是给程序阅读者看的,机器在编译和执行程序时,注释将被忽略。

2. C语言程序的书写规则

在书写C语言程序时,一般要注意以下几点:

(1) 在C语言中,虽然一行可写多个语句,一个语句也可占多行,但建议一行只写一个语句。

(2) 一般用适当的缩进格式书写程序,以提高程序的可读性和清晰性。

(3) 除非另有约定,C语言源代码一般用小写字母书写。

(4) 在程序代码中,应加上必要的注释。

1.3 C语言的基本词法

C语言作为计算机的一种程序设计语言,有自己的字符集、保留字和标识符及其构成规则,它们是C语言程序设计的基础。

1.3.1 字符集

字符是组成语言的最基本的元素。C语言字符集由字母、数字、空白符、图形符号、转义字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

① 字母:小写字母a~z共26个,大写字母A~Z共26个。

② 数字:0~9共10个。

③ 空白符:空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略。因此在程序中使用空白符与否,对程序的编译不发生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

④ 图形符号:表1.1列出了C语言允许使用的图形符号。

表 1.1 C语言图形符号表

符号	含义	符号	含义	符号	含义
~	波浪号	_	下划线	;	分号
!	感叹号	-	减号	"	双引号
@	at符号	+	加号	'	单引号
#	#号	=	等号	<	小于号

续表

符号	含义	符号	含义	符号	含义
\$	美元号		或符号	>	大于号
%	百分号	\	反斜线	,	逗号
~	异或号	{	左花括号	.	句号
&	与符号	}	右花括号	?	问号
*	星号	[	左方括号	/	正斜线
(	左圆括号	]	右方括号		空格
)	右圆括号	:	冒号	`	重音符号

⑤ 转义字符:转义字符是一种特殊的字符常量。转义字符以反斜线“\”开头,后跟一个或几个字符。转义字符具有特定的含义,不同于字符原有的意义,故称“转义”字符。例如,在前面各例题 printf 函数的格式串中用到的“\n”就是一个转义字符,其意义是“回车换行”。转义字符主要用来表示用一般字符不便于表示的控制代码,如表 1.2 所示列出了 C 语言常用的转义字符及其含义。

表 1.2 常用的转义字符及其含义

转义字符	转义字符的意义
\n	回车换行
\t	横向跳到下一个制表位
\v	竖向跳格
\b	退格
\r	回车
\f	换页
\\	反斜线字符“\”
\'	单引号符
\a	鸣铃
\ddd	1~3 位八进制数所代表的字符
\xhh	1~2 位十六进制所代表的字符

广义地讲,C 语言字符集中的任何一个字符均可用转义字符来表示。表中的 \ddd 和 \xhh 正是为此而提出的,ddd 和 xhh 分别为八进制的和十六进制的 ASCII 代码。如 \101 表示字母“A”,\102 表示字母“B”,\134 表示反斜线,\XOA 表示换行等,常用字符的 ASCII 码见附录 I。

【例 1.4】 分析下面程序,写出输出结果。

```
#include <stdio.h>
void main()
{
    int a,b,c;
    a=5;b=6;c=7;
    printf("%d\n\t%d %d\n %d %d\t\b%d\n",a,b,c,a,b,c);
}
```

说明:

程序在第一列输出 a 值 5 之后就是“\ n”,故回车换行;接着又是“\ t”,于是跳到下一制表位(设制表位间隔为 8),再输出 b 值 6;空两格再输出 c 值 7 后又是“\ n”,因此再回车换行;再空两格之后又输出 a 值 5;再空三格又输出 b 的值 6;在此后“\ t”跳到下一制表位(与上一行的 6 对齐),但下一转义字符“\ b”又使退回一格,故紧挨着 6 再输出 c 值 7。

程序运行后,屏幕显示:

```
5
    6 7
5    67
```

1.3.2 保留字

保留字是由 C 语言规定的具有特定意义的字符串,通常也称为关键字,C 语言的保留字只能小写,用户定义的标识符不应与保留字相同。如表 1.3 所示列出了 C 语言中可以使用的保留字。

表 1.3 C 语言中的保留字

关键字	含义	关键字	含义	关键字	含义
char	字符型	void	空值	while	while 循环
int	整型	const	常量	do	do 循环
long	长整型	volatile	可变量型	break	终止
short	短整型	auto	自动变量	continue	继续
float	单精度实型	extern	外部变量	goto	转向
double	双精度实型	static	静态变量	return	返回
unsigned	无符号型	register	寄存器	switch	开关语句
signed	有符号型	typedef	类型定义	default	默认
struct	结构体	if	if 语句	case	switch 的分支
union	共用体	else	else 语句	sizeof	返回字节数
enum	枚举	for	for 语句	include	预处理命令字

C 语言的保留字分为以下几类。

① 类型说明符:用于定义、说明变量、函数或其他数据结构的数据类型。如前面例题中用到的 int 等。

② 语句定义符:用于表示一个语句的功能。如例 1.3 中用到的 if else 就是条件语句的语句定义符。

③ 预处理命令字:用于表示一个预处理命令。如前面各例中用到的 include。

1.3.3 标识符及其构成规则

在程序中使用的变量名、函数名、标号等统称为标识符。除了库函数的函数名由系统定义外,其余都由用户自定义。C 语言规定,一个标识符必须符合下面所列的语法规则。

① 第一个字符必须是字母或下划线。



② 在第一个字母后面,可以是字母(A~Z,a~z)、数字(0~9)、下划线(_)组成的字符串。

③ 标识符的长度一般不超过 8 个字符。标准 C 语言不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某些版本中 C 语言规定标识符前 8 位有效,当两个标识符前 8 位相同时,则被认为是同一个标识符。

④ 在标识符中,大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

⑤ C 语言的关键字(保留字)不能作为标识符使用。

⑥ 标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名应尽量有相应的意义,以便阅读理解,做到“见名知义”。

【例 1.5】 判断以下标识符,哪些是合法的? 哪些是不合法的?

a, x, 3s, BOOK1, sum5, s * T, -3x, by-1, int, book1

根据构成标识符的语法规则。a, x, BOOK1, sum5, book1 是合法的,其中 BOOK1 和 book1 是有区别的,因为 C 语言区分大小写,它们是两个不同的标识符。

3s(标识符不允许以数字开头), s * T(* 不允许出现在标识符里), -3x(标识符不能以一开头), by-1(减号不允许出现在标识符里), int(关键字不允许作为标识符)是非法的。

1.4 C 程序的运行环境

1.4.1 C 语言程序实现过程

按照 C 语言语法规则编写的 C 程序称为源程序。设计好 C 的源程序后,要将它输入到计算机并得到最终结果,必须要经过编辑、编译、连接和运行 4 个环节,才能产生输出结果。其操作流程如图 1.1 所示。

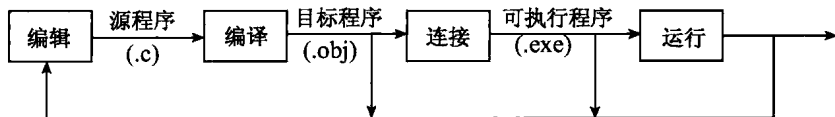


图 1.1 C 程序操作流程图

1. 编辑

编辑是在一定的环境下进行程序的输入和修改的过程。C 程序可以事先在纸上写好,也可以在编辑环境下直接输入计算机中。用某种计算机程序设计语言编写的程序称为源程序,保存后生成程序文件。C 源程序也可以使用各种编辑器进行编辑,比如 Turbo C 和 Visual C++ 这样的专用编辑系统。C 源程序在 Turbo C 2.0 环境下默认文件扩展名为“.c”,在 Visual C++ 6.0 环境下默认文件扩展名为“.cpp”。

2. 编译

编辑好的源程序不能直接被计算机所理解,源程序必须经过编译,生成计算机能够识别的机器代码。通过编译器将 C 源程序转换成二进制机器代码的过程称为编译,这些二进制机器代码称为目标程序,其扩展名为“.obj”。

编译阶段要进行词法分析和语法分析,这两种分析又称源程序分析。这一阶段主要是

分析程序的语法结构,检查C源程序的语法错误。如果分析过程中发现有不符合要求的语法,计算机就会及时报告给用户,将错误类型显示在屏幕上。

3. 连接

编译后生成的目标代码还不能直接在计算机上运行,其主要原因是编译器对每个源程序文件分别进行编译,如果一个程序有多个源程序文件,编译后这些源程序文件则可能分布在不同的地方。因此,需要把它们连接在一起,生成可以在计算机上运行的可执行文件。即使源程序仅由一个源文件构成,在源程序中,输入输出等标准函数不是用户自己编写的,而是直接调用系统库函数。因此,必须把目标程序与库函数进行连接。

连接工作一般由编译系统中的连接程序来完成,连接程序将由编译器生成的目标代码文件和库中的某些文件连接在一起,生成一个可执行文件。可执行文件的默认扩展名为“.exe”。

4. 运行

一个C源程序经过编译和连接后生成了可执行文件,可以在Windows环境下直接双击该文件运行程序,也可以在Visual C++ 6.0的集成开发环境下运行。

程序运行后,将在屏幕上显示运行结果或提示用户输入数据的信息。用户可以根据运行结果来判断程序是否有算法错误。在生成可执行文件之前,一定要保证编译和连接不出现错误和警告,这样才能正常运行。因为程序中有些警告虽然不影响生成可执行文件,但有可能导致结果错误的出现。

1.4.2 Visual C++ 6.0 上机环境介绍

VC++集成环境不仅支持C++程序的编译和运行,而且也支持C语言程序的编译和运行。通常VC++集成环境约定:当源程序文件的扩展名为.c时,编译和运行C程序,而当源程序文件的扩展名.cpp时,编译和运行C++程序。启动Visual C++ 6.0后,一般首先新建一个工程,再向工程中添加文件,然后编辑、编译、连接、运行源程序,最后查看程序运行结果。如果在创建文件前,没有创建相关工程,系统在编译时会提示是否要创建活动工程。

1. Visual C++ 6.0 的启动

单击【开始】|【程序】|【Microsoft Visual Studio 6.0】|【Microsoft Visual C++ 6.0】命令,可启动Visual C++ 6.0的集成开发环境,如图1.2所示。

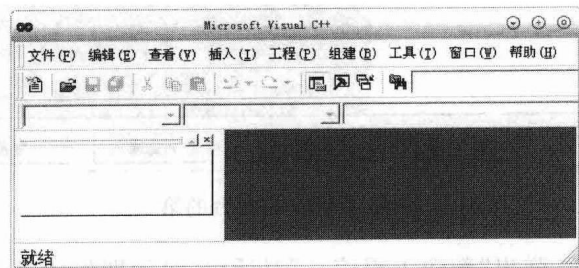


图 1.2 Microsoft Visual C++ 6.0 的集成开发环境

2. 新建工程

在“C:\student”目录下,新建一个名为“chapter01_01”的工程。新建工程 chapter01_01

的操作方法如下：

(1) 启动 Visual C++。

(2) 在 Visual C++ 主窗口中,单击【文件】|【新建】,打开“新建”对话框。

(3)在“新建”对话框中,单击选中“工程”标签下的“Win32 Console Application”(Win32 控制台应用程序)项。然后在“位置”文本框中指定新建工程的路径:“C:\STUDENT\chapter01_01”。最后在“工程”文本框中输入新建工程的名称:“chapter01_01”,如图 1.3 所示。

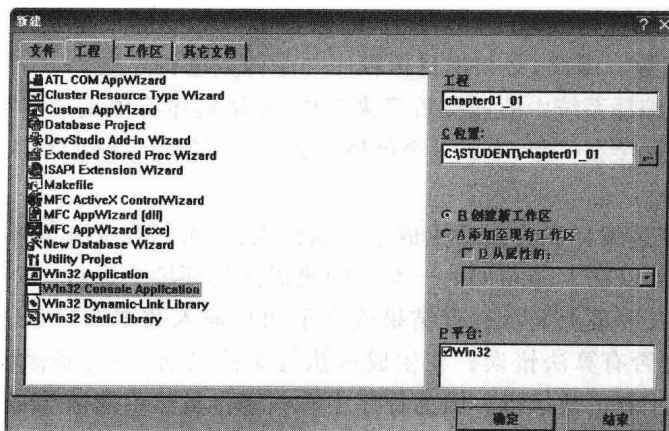


图 1.3 新建工程的“新建”对话框

(4)单击“确定”按钮,进入“Win32 Console Application - Step 1 of 1”对话框,选中“An empty project(一个空工程)”项,如图 1.4 所示。

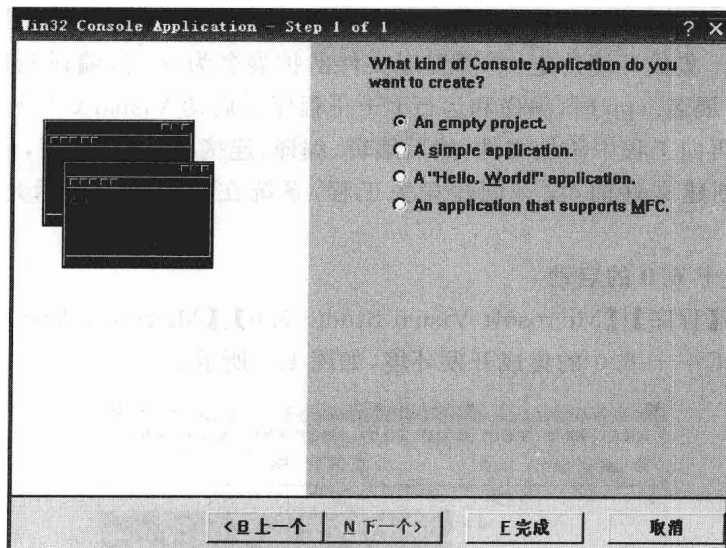


图 1.4 创建 Win32 控制台的第 1 步

(5)单击“完成”按钮,弹出“新建工程信息”对话框,显示即将新建的 Win32 控制台应用程序的框架说明,如图 1.5 所示。