

C语言

程序设计教程

基于Visual C++6.0环境

汤承林 姜仲秋 / 主编 张洪斌 / 主审

CYUYAN CHENGXU SHEJI JIAOCHENG

C 语言程序设计教程

——基于 Visual C ++ 6.0 环境

主 编 汤承林 姜仲秋
参 编 葛素娟 冯 钢
主 审 张洪斌

 天津大学出版社
TIANJIN UNIVERSITY PRESS

内容提要

本书紧贴全国计算机等级考试(二级 C)的考核内容,较全面地讲述了 C 语言程序设计的基本知识,主要内容包括概述、数据类型、运算符及表达式,基本输入/输出函数,选择结构,循环结构,数组,函数,指针,编译预处理命令,结构与共用,文件和图书管理系统案例介绍等,书后安排有与各章内容相关的 15 个实验。

本书章节例题有解题思路、参考代码、程序运行结果图、程序说明和注意点等。每一章都精选与全国计算机等级考试(二级 C)难度匹配的练习题,有助于读者复习、巩固所学知识。

本书结构严谨、重点突出、由浅入深、举例实用。本书的编程环境采用 Visual C++6.0,所有参考代码都在 Visual C++6.0 上调试通过。

本书可以作为各类高职高专、高等院校计算机专业及非计算机专业学习“计算机程序设计”的课程教材,也可作为参加全国计算机等级考试(二级 C)和广大计算机爱好者学习 C 语言程序设计的参考书。

图书在版编目(CIP)数据

C 语言程序设计教程:基于 Visual C++6.0 环境/汤承林,姜仲秋主编. —天津:天津大学出版社,2009. 8

ISBN 978-7-5618-3192-2

I. C… II. ①汤…②姜… III. C 语言 - 程序设计 - 高等学校 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 152435 号

出版发行 天津大学出版社
出 版 人 杨欢
地 址 天津市卫津路 92 号天津大学内(邮编:300072)
电 话 发行部:022-27403647 邮购部:022-27402742
网 址 www.tjup.com
印 刷 昌黎太阳红彩色印刷有限责任公司
经 销 全国各地新华书店
开 本 185 mm × 260 mm
印 张 22
字 数 550 千
版 次 2009 年 8 月第 1 版
印 次 2009 年 8 月第 1 次
定 价 36.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换

版权所有 侵权必究

前 言

C 语言是国内外目前广泛应用的程序设计语言。它功能强大,数据类型丰富,使用灵活,通用性强,并兼有面向硬件编程的低级语言特性和可读性强的高级语言特性。因此,C 语言已成为高校计算机专业和非计算机专业学生必修的一门计算机语言。

然而,在 C 语言的教学过程中,预期教学目标与最终教学效果往往有明显的差距,教师感觉难教,学生感觉难学、难理解,学了也不会编程。特别是高职高专院校的学生,学习时若过分注重选择适合高职高专学生学习特点的实用性 C 语言教材,学习后参加全国计算机等级考试(二级 C)时通过率很低;若学习时过分注重选择适合本科院校学生特点的学科性很强的 C 语言教材,理论性、学科性又太强,注重数据结构知识的讲解,这类教材适合参加全国计算机等级考试,但实用性差。在现时高职高专以“实用、够用”为度的情况下,教材以注重实用性为前提,必须改革教材内容,寻找能兼顾两者,既实用又具学科性的 C 语言教材。本教材就是在这种背景下,由具有多年高职高专 C 语言教学和开发经验的教师,经过认真分析高职高专学生学习特点和全国计算机等级考试(二级 C)的要求而编写的。

在本书编写过程中,主要从如下几个方面考虑。

(1) 充分考虑全国计算机等级考试要求,教材内容采用全国计算机等级考试(二级 C)考核内容。

(2) 教学内容的选择注重实用,选择易于被高职高专学生接受的例子。每个例子基本上分为解题思路、参考代码、解题说明、程序运行结果图和注意点。

(3) 以一个“图书管理系统”案例贯穿全书,以期使学生学习 C 语言后,能知道 C 语言到底能做什么,如何编写实用的 C 语言应用程序。“图书管理系统”案例根据需要分解在本书的部分章节中,针对章节所学内容,改造成利用所学章节内容编写的“图书管理系统”子模块。

(4) 充分考虑学生学习资料少、练习少、无适合全国等级考试练习题可做的窘境,在每章后均附有大量全国计算机等级考试练习题。习题主要分为选择题、填空题和编程题三类。

(5) 书后安排了 15 个实验,供学生上机练习。实验内容充分考虑到高职高专学生特点,把每个实验内容分为教师指导、练习、作业三部分。

本书的编程环境采用 Visual C++6.0,所有参考代码都在 Visual C++6.0 上调试通过。

全书由江苏淮安信息职业技术学院汤承林、姜仲秋副教授任主编,张洪斌研究员任主审。本书编写分工如下:第 3、4、6 和 10 章由姜仲秋编写;第 2、7、9 和 12 章由汤承林编写;第 5 和 8 章由葛素娟编写;第 1 和 11 章及实验部分由冯钢编写。本书在编写过程中得到了管曙亮高级工程师等人的帮助和支持,在此表示衷心感谢!

由于作者水平有限,加上时间仓促,错误之处在所难免,恳请广大读者批评指正。联系信箱:TCL@hcit.edu.cn 或 TCL12345678900@163.com

编 者
2009.5

目 录

第1章 概述	1
1.1 程序与算法	1
1.1.1 程序	1
1.1.2 算法	1
1.2 C语言的特点、源程序的书写格式	3
1.2.1 C语言的特点	3
1.2.2 C程序的构成	4
1.2.3 C语言程序的书写格式	4
1.2.4 C程序的三种基本结构和流程图	5
1.3 案例简介	6
1.4 Visual C++6.0上机操作	6
1.4.1 C程序编译与可执行文件的生成	6
1.4.2 Visual C++6.0的上机操作步骤	7
1.4.3 C程序调试	10
本章小结	13
习题一	13
第2章 数据类型、运算符及表达式	16
2.1 基本数据类型	16
2.2 常量与变量	17
2.2.1 常量与变量定义	17
2.2.2 整型数据	19
2.2.3 实型数据	22
2.2.4 字符型数据	23
2.2.5 字符串常量	25
2.3 数据类型转换	27
2.4 运算符及表达式	27
2.4.1 算术运算符与表达式	28
2.4.2 赋值运算符与表达式	30
2.4.3 逗号运算符	32
本章小结	32
习题二	33
第3章 基本输入/输出函数	37
3.1 输入/输出函数	37
3.1.1 格式化输入函数 scanf()	37

3.1.2	格式化输出函数 printf()	38
3.2	字符输入/输出函数	40
3.2.1	字符输入函数 getchar()	40
3.2.2	字符输出函数 putchar()	40
3.3	案例应用举例	41
	本章小结	42
	习题三	42
第4章	选择结构	48
4.1	关系运算和逻辑运算	48
4.1.1	关系运算符与表达式	48
4.1.2	逻辑运算符与表达式	49
4.2	if 语句	50
4.2.1	简单 if 语句	50
4.2.2	二分支 if 语句	51
4.2.3	二分支 if 语句嵌套	52
4.2.4	多分支 if 语句	53
4.2.5	条件运算符	55
4.3	switch 语句	55
4.4	综合实例	58
4.5	案例应用举例	62
	本章小结	63
	习题四	63
第5章	循环结构	70
5.1	goto 语句	70
5.2	for 语句	71
5.2.1	for 循环语句	71
5.2.2	for 循环嵌套	73
5.3	while 和 do-while 循环语句	76
5.3.1	while 循环语句	76
5.3.2	do-while 循环语句	78
5.3.3	for、while 和 do-while 循环嵌套	79
5.4	break 语句和 continue 语句	81
5.4.1	break 语句	81
5.4.2	continue 语句	82
5.5	综合实例	83
5.6	案例应用举例	85
	本章小结	86
	习题五	87
第6章	数组	98

6.1 一维数组	98
6.1.1 一维数组的定义及初始化	99
6.1.2 一维数组元素的引用	100
6.2 二维数组	105
6.2.1 二维数组的定义	105
6.2.2 二维数组的初始化	106
6.2.3 二维数组元素的引用	106
6.2.4 多维数组的定义	108
6.3 字符数组	109
6.3.1 字符数组的定义	109
6.3.2 字符数组的初始化	109
6.3.3 字符串与字符数组	110
6.3.4 字符串的输入/输出	111
6.3.5 字符串输入/输出函数	111
6.3.6 常用字符串处理函数	112
6.4 综合实例	114
6.5 案例应用举例	116
本章小结	118
习题六	118
第7章 函数	130
7.1 函数定义	130
7.2 函数的参数和函数的值	132
7.2.1 形式参数和实际参数	132
7.2.2 函数的返回值	133
7.2.3 函数的调用	134
7.2.4 函数原型	135
7.2.5 函数的调用实例	135
7.3 函数的嵌套调用	138
7.4 函数的递归调用	140
7.5 数组作为函数参数	141
7.5.1 数组元素作函数实参	142
7.5.2 数组名作为函数参数	143
7.6 变量的存储类别、作用域和生存期	146
7.6.1 局部变量与全局变量	146
7.6.2 变量的存储类别	149
7.7 内部函数与外部函数	152
7.7.1 内部函数	152
7.7.2 外部函数	152
7.8 案例应用举例	153

本章小结	154
习题七	154
第 8 章 指针	167
8.1 指针的概念和定义	167
8.1.1 地址和指针的概念	167
8.1.2 指针变量的定义	168
8.1.3 指针的操作	169
8.1.4 指向指针的指针	172
8.2 指针变量与函数参数	174
8.2.1 指针作为函数参数	174
8.2.2 值传递与地址传递的区别与联系	176
8.3 指针与数组	177
8.3.1 指向数组的指针	177
8.3.2 数组指针访问数组	178
8.3.3 指针变量访问数组	179
8.3.4 数组名作函数参数	180
8.4 指针与二维数组	182
8.4.1 二维数组的地址	182
8.4.2 指向由 n 个元素组成的一维数组的指针变量	184
8.4.3 指向二维数组元素的指针变量	186
8.5 字符串与指针	187
8.5.1 字符串的表示形式	187
8.5.2 字符数组与字符串指针作函数参数	188
8.6 指针数组	190
8.7 指针与函数	193
8.7.1 函数指针	193
8.7.2 返回值是指针的函数	195
8.8 main() 函数的参数	196
8.9 案例应用举例	197
本章小结	198
习题八	199
第 9 章 编译预处理命令	207
9.1 宏定义	207
9.1.1 不带参数的宏定义	207
9.1.2 带参数的宏定义	209
9.2 文件包含	210
9.3 条件编译	212
本章小结	215
习题九	215

第 10 章 结构与共用	219
10.1 结构与结构变量	219
10.1.1 结构的定义	219
10.1.2 结构变量的定义	220
10.1.3 结构变量的引用与赋值	222
10.1.4 结构变量的初始化	223
10.2 结构数组	224
10.2.1 结构数组的定义	224
10.2.2 结构数组的初始化	224
10.2.3 结构数组应用	225
10.3 指向结构的指针变量	226
10.3.1 指向结构变量的指针	226
10.3.2 指向结构数组的指针	228
10.3.3 指向结构变量的指针变量作函数参数	229
10.3.4 指向结构数组的指针变量作函数参数	230
10.4 类型定义符 typedef	232
10.5 结构与链表	233
10.5.1 动态存储结构	233
10.5.2 链表操作	235
10.6 共用	243
10.7 位运算	247
本章小结	250
习题十	250
第 11 章 文件	259
11.1 文件概述	259
11.1.1 文件及分类	259
11.1.2 文件类型指针	260
11.1.3 文件的打开与关闭	260
11.2 文本文件的读写	262
11.2.1 字符输入/输出函数 fgetc() 和 fputc() 及文件结束检测函数 feof()	262
11.2.2 字符串读写函数 fgets() 和 fputs()	265
11.3 二进制文件读写	266
11.4 文件的随机读写	269
11.4.1 文件定位	269
11.4.2 文件的随机读写	270
本章小结	271
习题十一	271
第 12 章 图书管理系统案例介绍	277
12.1 案例的目的和任务	277

12.2 案例知识要点综述	277
12.3 案例实训要求	277
12.4 案例需求分析	277
12.5 案例总体设计	277
12.6 案例详细设计	279
12.7 案例总结	300
实验	301
实验一 熟悉 Visual C ++6.0 环境	301
实验二 数据类型及运算符	302
实验三 输入/输出语句	303
实验四 选择结构	305
实验五 循环结构(一)	307
实验六 循环结构(二)	310
实验七 数组(一)	312
实验八 数组(二)	314
实验九 函数(一)	316
实验十 函数(二)	319
实验十一 指针(一)	322
实验十二 指针(二)	324
实验十三 结构与共用(一)	328
实验十四 结构与共用(二)	331
实验十五 文件	334
附录 1 ASCII 表	337
附录 2 C 库函数所在头文件	338
参考文献	339

第 1 章 概述

C 语言的设计始于 1971 年,后经多次改进,迅速成为最受欢迎的计算机编程语言之一。许多流行软件都是用 C 语言编写的。

C 语言的编程环境有 Turbo C、Borland C、Visual C++ 等,它们都内嵌了 C 语言编译程序。

本章介绍 C 语言程序的基本知识:

- (1) 算法、C 语言的特点、C 语言的构成、书写格式介绍;
- (2) 本书案例“图书管理系统”简介;
- (3) Visual C++6.0 上机操作步骤介绍。

1.1 程序与算法

1.1.1 程序

“程序”一词来自生活,通常指完成某件事情的一种既定的方式和过程。可以把程序看成对一系列动作的执行过程的描述。例如,学生去食堂吃饭的行为可以描述为:

- (1) 带上饭碗去食堂;
- (2) 排队;
- (3) 打饭;
- (4) 刷卡;
- (5) 就餐;
- (6) 洗碗;
- (7) 离开食堂。

这是个生活中的程序性活动的例子,计算机完成某件事情与生活中解决某个问题的过程非常相似。为了完成人们交给计算机的问题,计算机提供了一套指令,其中每一条指令对应着计算机能执行的一个基本动作,为了让计算机解决某个问题而编写的逐条执行的指令序列就称为程序。

1.1.2 算法

为解决一个问题而采取的方法与步骤,称为“算法”(algorithm)。或者说,算法是解题方法的精确描述。解决一个问题的过程就是实现一个算法的过程。对同一个问题,往往有不同的解题方法,各种方法的执行速度可能不同;为了有效地执行程序,应当选择适当的算法。选择算法的主要标准是考虑它的正确性、可靠性、简单性和易理解性,其次是考虑所需要的存储空间和执行速度等。

一个程序应包括对数据的描述和对操作的描述。

- (1) 对数据的描述。在程序中要指定数据的类型和组织形式,即数据结构(data struc-

ture)。

(2)对操作的描述。即操作步骤,也就是算法。著名计算机科学家 Nikiklaus Wirth 对程序提出如下公式:

$$\text{程序} = \text{数据结构} + \text{算法}$$

一个算法应具有以下特点。

- (1)有穷性。一个算法应包含有限的操作步骤,而不能是有限的。
- (2)确定性。算法中每一个步骤应当是确定的,而不能是含糊的、模棱两可的。
- (3)有零个或多个输入。所谓输入是指执行指定的算法时,需要外界提供输入信息。
- (4)有一个或多个输出。
- (5)有效性。算法中每一个步骤应当能有效地执行,并得到确定的结果。


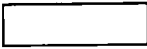
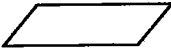



对于程序设计人员,必须会设计算法,并根据算法写出程序。

【例 1.1】求 $1+2+3+\dots+100$ 。假设 SUM 代表累加和, N 代表累加数个数。其程序如下。

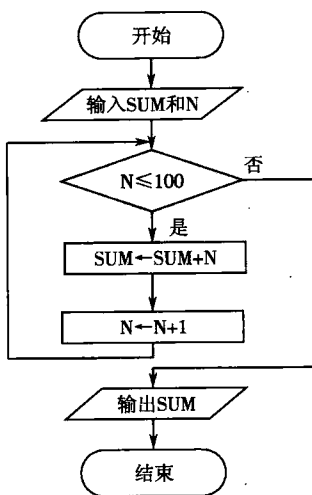
- (1)SUM = 0, N = 1。
 - (2)求 SUM + N, 得到的结果放在 SUM 中。
 - (3)N 加 1 后放在 N 中。
 - (4)如果 $N \leq 100$, 跳转到(2); 如果 $N > 100$, 则算法结束。此时 SUM 的值就是所求结果。
- 算法通常按某种规则来描述,常用的有自然语言、传统流程图、N-S 结构流程等。

流程图是算法的一种图形化表示方式,它直观、清晰,有利于人们设计算法。传统流程图常用的符号如表 1.1 所示。

表 1.1 传统流程图常用符号

符号	描述
	程序开始或结束
	计算机步骤/处理符号
	输入/输出指令
	判断和分支
	连接符
	流程线

前面给出的计算 1 到 100 之和的算法用流程图表示如图 1.1 所示。

图 1.1 $1 + 2 + \dots + 100$ 流程图

1.2 C 语言的特点、源程序的书写格式

1.2.1 C 语言的特点

C 语言是现今最流行的几种程序设计语言之一。与其他语言相比, C 语言具有如下主要特点。

(1) 语言简洁、紧凑, 使用方便、灵活。ANSI C 一共只有 32 个关键字, 如表 1.2 所示。

表 1.2 ANSI C 的 32 个关键字

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

(2) 运算符丰富。

(3) 数据结构丰富, 具有现代化语言的各种数据结构。

(4) 具有结构化的控制语句。

(5) 语法限制不太严格, 程序设计自由度大。

(6) 允许直接对内存物理地址的数据进行操作, 可以实现汇编语言的大部分功能, 同时具有高级语言的结构化和可读性、可移植性等优点。有人把 C 语言称为“高级语言中的低级语言”或“中级语言”, 但一般仍习惯将 C 语言称为高级语言, 因为 C 程序也要通过编译、连接才能得到可执行的目标程序。

(7)生成目标代码运行效率高。

1.2.2 C 程序的构成

例 1.2 展示 C 程序的构成与书写格式。

【例 1.2】求 $1+2+3+\dots+100$ 。假设 SUM 代表累加和,N 代表累加和个数。

```
#include <stdio.h>           /* 标准输入/输出头文件 */
void main()                 /* 主函数 */
{
    int SUM,N;              /* 定义变量 */
    SUM=0;                  /* 给变量赋值 */
    N=1;
    while(N <= 100)
    {
        SUM = SUM + N;
        N = N + 1;
    }
    printf("SUM = %d\n",SUM); //输出结果
}
```

程序分析如下。

(1)所有 C 语言程序必须包括一个函数 main(),表示主函数。它是 C 语言程序的入口。函数 main()后面由一对花括号{ }括起来的部分称为程序的主体,程序从函数 main()的第 1 个可执行语句开始执行。

(2)/*……*/和//……表示注释部分,用于对程序进行说明,增强代码的可读性,但对程序编译与运行不起作用。前者表示块注释,后者表示单行注释。语句的注释可以有 多行,可以放在程序的任何地方,但不可插入到关键字或标识符中间,否则编译时会出错。

(3)printf()是 C 语言程序中标准库函数的输出函数,其功能是输出双引号内的字符串,而“\n”是换行控制符,即输出完毕后自动换行。由于 printf()是 C 语言程序中的库函数,所以在程序的开始必须引用库函数的接口说明文件 stdio.h,此文件中包含函数 printf()的原型说明。

(4)主函数前的关键字 void 表示该函数不返回任何结果,缺省时表示返回整型变量值。

1.2.3 C 语言程序的书写格式

根据例 1.2,可以总结 C 语言程序的书写格式如下。

(1)C 程序采用块注释。块注释书写格式为:

```
/* 注释部分 */
```

注释只是增强程序的可读性,不参加程序的编译与执行。书写时要注意“/”与“*”之间不能含有空格。C 程序在 Visual C ++6.0 编程环境中,也可以采用 C ++的注释方法,即要对某行进行注释,只需在该行后面加上“//”并标明注释部分即可。

(2)C 语言中,标识符是一个名字,一般采用小写字母作标识符。C 语言允许用作标识的字符有以下几种。

①26个英文字母,包括大小写(共52个)。

②数字0、1、……、9。

③下画线“_”。

④C语言标识符由英文字母、数字和下画线组成,但不能以数字打头,标识符之间不能含有空格,长度为1~32。

(3)C语言区分大小写。如SUM、sum和Sum表示三个不同的标识符。

(4)C程序书写格式灵活,一个语句可连续写在多行上,一行中也可以写多个语句。如例1.2中的“SUM=0;N=1;”两个语句可以写在一行上,也可以写在两行上。

(5)为了使书写的程序结构清晰、层次分明,建议采用“缩进对齐”的格式编辑C语言源程序。

1.2.4 C程序的三种基本结构和流程图

C语言程序有以下三种基本结构。

(1)顺序结构。如图1.2所示,虚线框内是一个顺序结构,其中A和B两个框是顺序执行的。顺序结构是最简单的基本结构。

(2)选择结构。如图1.3所示,虚线框内是一个选择结构。此结构包括一个判断框,根据给定的条件P是否成立而选择执行A框或B框。

(3)循环结构。循环结构是反复执行某一部分的操作。循环结构有以下两种。

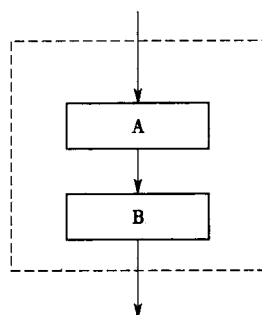


图1.2 顺序结构

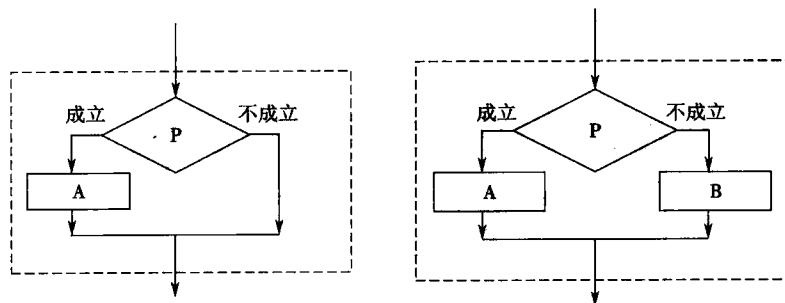


图1.3 选择结构

①当型循环结构。如图1.4(a),当条件p1成立时才执行A框操作,否则A框操作一次也不执行。

②直到型循环结构。如图1.4(b),必须执行A框一次后,才去判定条件p2是否成立,成立继续执行A框操作,否则A框操作不执行。

三种基本结构具有以下共同特点:

- (1)只有一个入口;
- (2)只有一个出口;
- (3)结构内的每一部分都有机会被执行;
- (4)结构内不能存在“死循环”。

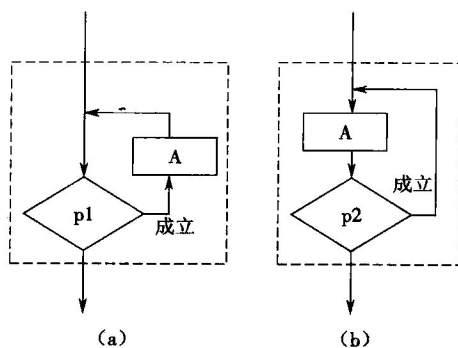


图 1.4 循环结构

(a) 当型循环结构; (b) 直到型循环结构

1.3 案例简介

本书在部分章节中抽取或改造了书后案例部分程序,以促进读者理解与掌握 C 语言的学习。许多读者不知道学习 C 语言到底应用在何处,更不知道如何编写一个完整的 C 程序的方法和步骤,本书第 12 章给出一个完整的“图书管理系统”的 C 语言程序源代码。“图书管理系统”共分三个模块:

- (1) 图书信息的增加、删除、修改和查询;
- (2) 借书卡信息的增加、删除、修改和查询;
- (3) 图书的借还与借还信息查询。

其主菜单如图 1.5 所示。

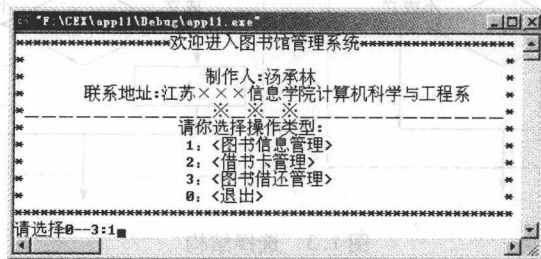


图 1.5 本书案例程序主菜单

1.4 Visual C++ 6.0 上机操作

1.4.1 C 程序编译与可执行文件的生成

C 语言是高级程序语言,用 C 语言写出的程序通常称为 C 源程序,人们容易使用、书写和阅读 C 源程序,但计算机却不能直接执行,因为计算机只能识别和执行特定的二进制形式机器语言程序。为使计算机能完成某个 C 源程序所描述的工作,就必须把 C 源程序转换成计算

机能够识别和执行的二进制机器语言。C 语言程序编译和执行过程如图 1.6 所示。

C 语言程序可执行文件生成过程如下。

(1) 利用编辑器编写生成 C 语言源程序(文本文件)。编辑器可用记事本、Word 等。

(2) 采用 C 编译器将源程序编译成二进制的目标文件(伪代码),生成的目标文件扩展名为 .obj。

(3) 采用 C 连接程序将目标文件(*.obj)与库文件(*.h)连接,生成可执行文件。可执行文件扩展名为 .exe,可执行文件可脱离 C 环境运行。

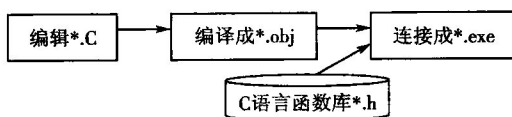


图 1.6 C 源程序转换成可执行文件

1.4.2 Visual C ++ 6.0 的上机操作步骤

全国计算机等级考试(二级 C)采用 Visual C ++ 6.0 环境。为了与全国计算机等级考试环境相匹配,本书的所有源程序皆用 Visual C ++ 6.0 编辑、编译与运行。

用 Visual C ++ 6.0 创建的 C 程序被存储为一个独立工程(Project),每个工程会新建一个文件夹,工程中含有一组文件,这组文件可以具有不同的扩展名,其中部分文件由 Visual C ++ 6.0 自动创建,这组文件组合在一起形成一个完整的应用程序。

下面以例 1.2 为例(例 1.2 的源程序文件名为 ex1_2.c)介绍 Visual C ++ 6.0 中建立工程并进行 C 程序调试的主要操作步骤。

1. 启动 Visual C ++ 6.0

如图 1.7 所示,从“开始”按钮中的“程序”项中启动“Microsoft Visual C ++ 6.0”。

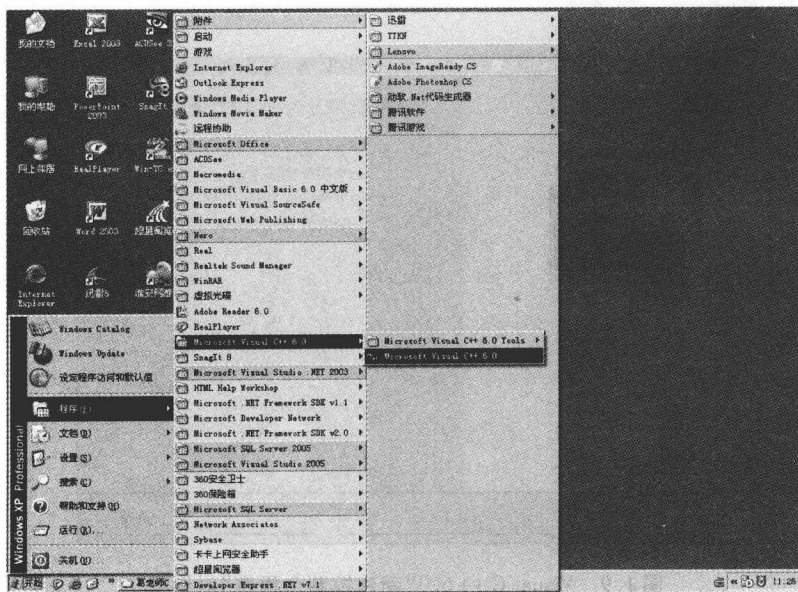


图 1.7 启动 Visual C ++ 6.0

2. 进入 Visual C ++ 6.0 环境

执行“文件”→“新建”菜单项,弹出“新建”对话框,如图 1.8 所示。该对话框有分别用于