

高等学校规划教材·计算机实用软件应用系列教程

# C 语言 程序设计实例教程

主编 曹岩



西北工业大学出版社

高等学校规划教材·计算机实用软件应用系列教程

# C 语言程序设计实例教程

主编 曹岩

编者 马翔 李建英 张文琪

图书在版编目(CIP)数据

C语言程序设计实例教程 / 曹岩主编. —西安: 西北工业大学出版社, 2010.8.

ISBN 978-7-5612-2897-7

I. ①C… II. ①曹… III. ①C语言—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第123456号

出版发行: 西北工业大学出版社  
通信地址: 西安市太白南路2号  
电 话: (029) 88493000  
网 址: www.nwpu.com  
电子邮箱: computer@nwpu.com  
印刷: 陕西兴平报社印刷厂  
开 本: 787 mm × 1 092 mm 1/16  
印 张: 13.2  
字 数: 328千字  
定 次: 2010年8月第1版  
定 价: 28.00元

西北工业大学出版社

**【内容简介】**本书通过实例详细系统地介绍了 C 语言程序设计的方法和过程。主要内容包括：VC 的安装及使用，格式输入与输出，字符及字符串数据的输入与输出，运算符的优先级和结合性，选择结构，循环结构，数组，字符数组，函数与函数调用，变量的作用域规则，结构体、共用体、枚举，指针与数组、结构体、字符串，预处理命令，申请内存空间，键盘输入及磁盘输出，给自己的程序加上行号，文件存储学生管理系统，文件分割，文件合并，文件加密，链表，栈，工程文件组织，预编译，库等。

本书内容全面实用，实例丰富，可作为各类高等学校教材或教学参考书，也可供科学与工程计算、数值计算、图形处理、自动控制、科学绘图、数据分析、设计、制造、控制、机械、电子、电器等领域从事 C 语言程序设计的科学研究和工程技术的人员参阅。

#### 图书在版编目 (CIP) 数据

C 语言程序设计实例教程/曹岩主编. —西安: 西北工业大学出版社, 2010.8  
高等学校规划教材·计算机实用软件应用系列教程  
ISBN 978-7-5612-2864-7

I. ①C… II. ①曹… 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 158343 号

出版发行: 西北工业大学出版社

通信地址: 西安市友谊西路 127 号 邮编: 710072

电 话: (029) 88493844 88491757

网 址: www.nwpup.com

电子邮箱: computer@nwpup.com

印 刷 者: 陕西兴平报社印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 13.5

字 数: 358 千字

版 次: 2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

定 价: 26.00 元

# 前 言

C 语言作为一种结构化的计算机程序设计语言,是目前最流行和使用最广泛的计算机语言之一,以其强大的功能在系统软件开发、科学计算、自动控制等各个领域有着十分广泛的应用。C 语言具有表达能力强,概念和功能丰富,目标代码质量高,使用灵活方便,可移植性好等特点。C 语言既具有高级语言的优点,又具有低级语言的特点,特别适合于编写系统软件和嵌入式软件。

C 语言是面向过程的高级程序设计语言,而高级程序设计语言是当今大学生的必修课程,在绝大多数高等院校几乎所有的专业都把 C 语言作为第一门程序设计语言课程。全国计算机等级考试、全国计算机应用技术证书考试也都将 C 语言列入了考试范围。尽管 C 语言具有其他高级程序设计语言不可比拟的优点,但是 C 语言的概念多,规则复杂,知识丰富,学习量大,容易出错。学生往往感觉困难,普遍反映难学,难懂,理论与实践脱节。

本书从使用者的角度出发,通过集经验技巧于一体的典型实例讲解,系统介绍了 C 语言程序设计的方法和过程。全书共包括 25 章:第 1 章 VC 的安装及使用,第 2 章格式输入与输出,第 3 章字符及字符串数据的输入与输出,第 4 章运算符的优先级和结合性,第 5 章选择结构,第 6 章循环结构,第 7 章数组,第 8 章字符数组,第 9 章函数与函数调用,第 10 章变量的作用域规则,第 11 章结构体、共用体、枚举,第 12 章指针与数组、结构体、字符串,第 13 章预处理命令,第 14 章申请内存空间,第 15 章键盘输入磁盘输出,第 16 章给自己的程序加上行号,第 17 章文件存储学生管理系统,第 18 章文件分割,第 19 章文件合并,第 20 章文件加密,第 21 章链表,第 22 章栈,第 23 章工程文件组织,第 24 章预编译,第 25 章库。

本书内容全面实用,实例丰富,采用循序渐进、通俗易懂的讲解方法以及理论与实践相结合的原则,在阐明 C 语言的基本概念、语法与正确使用的基础上,通过一系列经典实例系统地讲述 C 语言程序设计的基本方法、规则和技巧,使学生掌握利用 C 语言进行结构化程序设计的技术和方法,重在培养学生的实际操作能力和程序设计能力。每一章首先给出 C 语言程序设计的学习内容,然后通过一系列的实例进行详细阐述,每一程序设计实例都给出具体算法描述和运行结果,以加深对知识点的理解,提高分析问题和设计算法的能力,对于容易出错之处也给出应注意的事项,从而达到提高学生运用所学知识解决实际问题的能力。

全书由曹岩主编,马翔、李建英、张文琪编写。其中,第 1, 14, 23~25 章由西安电



# 目 录

## 第 1 章 VC 6.0 的安装与使用 ..... 1

### 1.1 引言 ..... 1

### 1.2 VC 6.0 的安装 ..... 1

### 1.3 VC 6.0 的使用 ..... 5

#### 1.3.1 编译.c 文件 ..... 5

#### 1.3.2 调试 ..... 8

## 第 2 章 格式输入与输出 ..... 13

### 2.1 基础知识 ..... 13

### 2.2 编程思路 ..... 13

### 2.3 实例源代码 ..... 15

#### 2.3.1 实例 2-1: 练习 int 类型和 long int 类型的各种输出方法 ..... 15

#### 2.3.2 实例 2-2: 练习浮点类型的各种输出方法 ..... 16

### 2.4 小贴士 ..... 17

## 第 3 章 字符及字符串数据的输入与输出 ..... 18

### 3.1 基础知识 ..... 18

### 3.2 编程思路 ..... 20

### 3.3 实例源代码 ..... 20

#### 3.3.1 实例 3-1: putchar 函数和 getchar 函数使用 ..... 20

#### 3.3.2 实例 3-2: puts 函数和 gets 函数使用 ..... 21

#### 3.3.3 实例 3-3: c 格式使用 ..... 22

#### 3.3.4 实例 3-4: s 格式使用 ..... 22

#### 3.3.5 实例 3-5: 各种字符串输入和输出函数的比较 ..... 23

### 3.4 小贴士 ..... 25

## 第 4 章 运算符的优先级和结合性 ..... 26

### 4.1 基础知识 ..... 26

### 4.2 编程思路 ..... 28

### 4.3 实例源代码 ..... 28

#### 4.3.1 实例 4-1: 赋值语句结合各类运算符的使用 ..... 28

#### 4.3.2 实例 4-2: 各数值类型在内存中所占字节数 ..... 30

#### 4.3.3 实例 4-3: 使用异或运算数据交换 ..... 30

#### 4.3.4 实例 4-4: 使用位运算符转换 IP 地址 ..... 31

#### 4.3.5 实例 4-5: 使用递增操作符 ..... 32

### 4.4 小贴士 ..... 32

## 第 5 章 选择结构 ..... 34

### 5.1 基础知识 ..... 34

### 5.2 编程思路 ..... 35

### 5.3 实例源代码 ..... 35

#### 5.3.1 实例 5-1: 猜数程序 ..... 35

#### 5.3.2 实例 5-2: 判断闰年 ..... 37

#### 5.3.3 实例 5-3: 输入某年某月某日, 判断这一天是这一年的第几天 ..... 38

#### 5.3.4 实例 5-4: 输入一个字符, 判断其是否大写字母, 如果是, 将它转换成小写 ..... 39

#### 5.3.5 实例 5-5: 求解一元二次方程 ..... 40

### 5.4 小贴士 ..... 41

## 第 6 章 循环结构 ..... 42

### 6.1 基础知识 ..... 42

### 6.2 编程思路 ..... 43

### 6.3 实例源代码 ..... 43

#### 6.3.1 实例 6-1: 求 Fibonacci 数列前 40 个数 ..... 43

#### 6.3.2 实例 6-2: 判断 m 是否是素数 ..... 44

6.3.3 实例 6-3: 输入两个正整数 m 和 n, 求其最大公约数和最小公倍数 .....	45	9.3.5 实例 9-5: Hanoi 塔 .....	78
6.3.4 实例 6-4: 输入一行字符, 分别统计出其中英文字母、空格、数字和其他字符的个数 .....	46	9.4 小贴士 .....	79
6.3.5 实例 6-5: 打印菱形图案 .....	47	<b>第 10 章 变量的作用域规则</b> .....	80
6.4 小贴士 .....	48	10.1 基础知识 .....	80
<b>第 7 章 数组</b> .....	49	10.2 编程思路 .....	82
7.1 基础知识 .....	49	10.3 实例源代码 .....	83
7.2 编程思路 .....	51	10.3.1 实例 10-1: auto 定义变量的用法 .....	83
7.3 实例源代码 .....	51	10.3.2 实例 10-2: static 定义静态局部变量的用法 .....	84
7.3.1 实例 7-1: 求一定范围内的素数的个数 .....	51	10.3.3 实例 10-3: 学习使用 static 的另一用法 .....	85
7.3.2 实例 7-2: 冒泡排序 .....	52	10.3.4 实例 10-4: 学习使用 external 的用法 .....	85
7.3.3 实例 7-3: 杨辉三角形 .....	54	10.3.5 实例 10-5: 学习使用 register 定义变量的方法 .....	86
7.3.4 实例 7-4: 数字出局游戏 .....	55	10.3.6 实例 10-6: 内部函数和外部函数的使用方法 .....	86
7.3.5 实例 7-5: 魔方阵 .....	56	10.4 小贴士 .....	88
7.3.6 实例 7-6: 统计班级成绩 .....	58	<b>第 11 章 结构体、共用体、枚举</b> .....	89
7.4 小贴士 .....	59	11.1 基础知识 .....	89
<b>第 8 章 字符数组</b> .....	60	11.2 编程思路 .....	92
8.1 基础知识 .....	60	11.3 实例源代码 .....	93
8.2 编程思路 .....	61	11.3.1 实例 11-1: 学生成绩管理 .....	93
8.3 实例源代码 .....	62	11.3.2 实例 11-2: 人员信息管理 .....	95
8.3.1 实例 8-1: 统计单词个数 .....	62	11.3.3 实例 11-3: 枚举类型的综合运用 .....	98
8.3.2 实例 8-2: 扑克牌数据模拟 .....	63	11.3.4 实例 11-4: 计算日期是本年的第几天 .....	99
8.3.3 实例 8-3: 字符比较 .....	64	11.4 小贴士 .....	101
8.3.4 实例 8-4: 两个字符串的连接 .....	66	<b>第 12 章 指针与数组、结构体、字符串</b> .....	102
8.4 小贴士 .....	67	12.1 基础知识 .....	102
<b>第 9 章 函数与函数调用</b> .....	68	12.2 编程思路 .....	105
9.1 基础知识 .....	68	12.3 实例源代码 .....	106
9.2 编程思路 .....	70	12.3.1 实例 12-1: 交换两个数 .....	106
9.3 实例源代码 .....	71		
9.3.1 实例 9-1: 求最大值和最小值 .....	71		
9.3.2 实例 9-2: 求一元二次方程的根 .....	73		
9.3.3 实例 9-3: 提取最长的单词 .....	75		
9.3.4 实例 9-4: 求阶乘 .....	76		

12.3.2 实例 12-2: 选择排序.....	107	17.4 实例源代码.....	137
12.3.3 实例 12-3: 深入理解指针与 数组.....	109	17.5 小贴士.....	138
12.3.4 实例 12-4: 指针与字符串.....	110	<b>第 18 章 文件分割</b> .....	139
12.3.5 实例 12-5: 打印学生成绩.....	113	18.1 背景知识.....	139
12.3.6 实例 12-6: 数值比较.....	114	18.2 基础知识.....	139
12.4 小贴士.....	116	18.3 编程思路.....	139
<b>第 13 章 预处理命令</b> .....	117	18.4 实例源代码.....	139
13.1 基础知识.....	117	18.5 小贴士.....	141
13.2 编程思路.....	118	<b>第 19 章 文件合并</b> .....	142
13.3 实例源代码.....	119	19.1 背景知识.....	142
13.3.1 实例 13-1: 宏与常量.....	119	19.2 编程思路.....	142
13.3.2 实例 13-2: 电报密码.....	120	19.3 实例源代码.....	142
13.4 小贴士.....	121	19.4 小贴士.....	144
<b>第 14 章 申请内存空间</b> .....	122	<b>第 20 章 文件加密</b> .....	145
14.1 背景知识.....	122	20.1 背景知识.....	145
14.2 基础知识.....	122	20.2 基础知识.....	145
14.3 实例源代码.....	123	20.3 编程思路.....	145
<b>第 15 章 键盘输入磁盘输出</b> .....	129	20.4 实例源代码.....	146
15.1 背景知识.....	129	20.5 小贴士.....	147
15.2 基础知识.....	129	<b>第 21 章 链表</b> .....	148
15.3 编程思路.....	131	21.1 背景知识.....	148
15.4 实例源代码.....	131	21.1.1 抽象数据类型.....	148
15.5 小贴士.....	132	21.1.2 线性表.....	148
<b>第 16 章 给自己的程序加上行号</b> .....	133	21.1.3 线性链表.....	149
16.1 背景知识.....	133	21.2 基础知识.....	153
16.2 基础知识.....	133	21.3 编程思路.....	153
16.3 编程思路.....	134	21.4 实例源代码.....	153
16.4 实例源代码.....	134	21.5 小贴士.....	157
16.5 小贴士.....	135	<b>第 22 章 栈</b> .....	159
<b>第 17 章 文件存储学生管理系统</b> .....	136	22.1 背景知识.....	159
17.1 背景知识.....	136	22.2 基础知识.....	161
17.2 基础知识.....	136	22.3 编程思路.....	162
17.3 编程思路.....	137	22.4 实例源代码.....	162
		22.5 小贴士.....	165

第 23 章 工程文件组织 ..... 166

- 23.1 工程概念 ..... 166
- 23.2 工程文件编写 ..... 166
- 23.3 构建一个控制台应用程序 ..... 167
- 23.4 Makefile ..... 172
- 23.5 多工程调试 ..... 172

第 24 章 预编译 ..... 175

- 24.1 背景知识 ..... 175
  - 24.1.1 文件包含 ..... 175
  - 24.1.2 宏定义 ..... 175
  - 24.1.3 条件编译 ..... 176
- 24.2 基础知识及编程思路 ..... 176
  - 24.2.1 文件包含 ..... 176

24.2.2 宏定义 ..... 179

24.2.3 条件编译 ..... 180

24.3 实例源代码 ..... 181

第 25 章 库 ..... 188

- 25.1 背景知识 ..... 188
- 25.2 Jpeg 压缩库介绍 ..... 188
- 25.3 库的建立 ..... 189
  - 25.3.1 静态链接库 ..... 189
  - 25.3.2 动态链接库 ..... 191
- 25.4 库的使用 ..... 194
  - 25.4.1 静态链接库的使用 ..... 194
  - 25.4.2 动态链接库的使用 ..... 206

第 26 章 字符串 ..... 201

- 26.1 背景知识 ..... 201
  - 26.1.1 字符串 ..... 201
  - 26.1.2 字符串常量 ..... 202
  - 26.1.3 字符串数组 ..... 202
  - 26.1.4 字符串指针 ..... 202
- 26.2 基础知识及编程思路 ..... 202
  - 26.2.1 字符串 ..... 202

第 27 章 数组 ..... 211

- 27.1 背景知识 ..... 211
  - 27.1.1 数组 ..... 211
  - 27.1.2 数组的声明 ..... 211
  - 27.1.3 数组的初始化 ..... 211
  - 27.1.4 字符串数组 ..... 211
- 27.2 基础知识及编程思路 ..... 211
  - 27.2.1 数组 ..... 211

第 28 章 指针 ..... 213

- 28.1 背景知识 ..... 213
  - 28.1.1 指针 ..... 213
  - 28.1.2 指针的声明 ..... 213
  - 28.1.3 指针的初始化 ..... 213
  - 28.1.4 指针的运算 ..... 213
  - 28.1.5 指针与数组 ..... 213
  - 28.1.6 指针与字符串 ..... 213
  - 28.1.7 指针与函数 ..... 213
  - 28.1.8 指针与结构体 ..... 213
  - 28.1.9 指针与联合体 ..... 213
  - 28.1.10 指针与枚举 ..... 213
  - 28.1.11 指针与文件 ..... 213
- 28.2 基础知识及编程思路 ..... 213
  - 28.2.1 指针 ..... 213

第 29 章 结构体 ..... 213

- 29.1 背景知识 ..... 213
  - 29.1.1 结构体 ..... 213
  - 29.1.2 结构体的声明 ..... 213
  - 29.1.3 结构体的初始化 ..... 213
  - 29.1.4 结构体的运算 ..... 213
  - 29.1.5 结构体与数组 ..... 213
  - 29.1.6 结构体与字符串 ..... 213
  - 29.1.7 结构体与函数 ..... 213
  - 29.1.8 结构体与文件 ..... 213
  - 29.1.9 结构体与枚举 ..... 213
  - 29.1.10 结构体与联合体 ..... 213
- 29.2 基础知识及编程思路 ..... 213
  - 29.2.1 结构体 ..... 213

第 30 章 枚举 ..... 213

- 30.1 背景知识 ..... 213
  - 30.1.1 枚举 ..... 213
  - 30.1.2 枚举的声明 ..... 213
  - 30.1.3 枚举的初始化 ..... 213
  - 30.1.4 枚举的运算 ..... 213
- 30.2 基础知识及编程思路 ..... 213
  - 30.2.1 枚举 ..... 213

第 31 章 联合体 ..... 213

- 31.1 背景知识 ..... 213
  - 31.1.1 联合体 ..... 213
  - 31.1.2 联合体的声明 ..... 213
  - 31.1.3 联合体的初始化 ..... 213
  - 31.1.4 联合体的运算 ..... 213
- 31.2 基础知识及编程思路 ..... 213
  - 31.2.1 联合体 ..... 213

第 32 章 文件 ..... 213

- 32.1 背景知识 ..... 213
  - 32.1.1 文件 ..... 213
  - 32.1.2 文件的声明 ..... 213
  - 32.1.3 文件的初始化 ..... 213
  - 32.1.4 文件的运算 ..... 213
- 32.2 基础知识及编程思路 ..... 213
  - 32.2.1 文件 ..... 213

第 33 章 附录 ..... 213

- 33.1 附录 A ..... 213
- 33.2 附录 B ..... 213
- 33.3 附录 C ..... 213

# 第 1 章 VC 6.0 的安装与使用

## 1.1 引言


本书是通过列举实例为读者讲解 C 语言的使用方法，故要选择一款合适的 C 语言开发环境，这里选择了 VC 6.0，因为其操作简便、界面简洁、功能强大，尤其适合 C 语言基础的学习，更重要的是 VC 6.0 的 C 编译器对标准 C 的支持几乎达到了完美的水平。现如今，编译环境层出不穷，而且功能更为强大，但是，基础并没有变，所不同的是这些环境更为支持诸如窗口程序、网页程序等的编写。凡事都应从浅入深，C 语言被誉为高级编程语言的基础，先从基础 C 开始，再去学习和应用更高级的 C 系列语言和环境，才能理解语言的精髓，写出更好的程序。更何况，在嵌入式等开发环境下，基础 C 的地位还无法替代，VC 6.0 依然是绝大多数程序员开发 C 程序的首选。本章介绍 VC 6.0 的安装与使用。

## 1.2 VC 6.0 的安装

在这里以图示的方法详细介绍其安装过程。首先需要下载 VS 6.0 的安装包，VC 6.0 包含于其中，网上有很多资源，这里不多阐述，安装包如图 1-1 所示。



图 1-1 VS 6.0 安装包

用鼠标双击图标 ，进入安装界面，如图 1-2 所示。

用鼠标单击“Next”，进入如图 1-3 所示界面。这一界面为安装协议，在安装时应仔细阅读协议，如果确认无误选中“ I accept the agreement”复选框，并用鼠标单击“Next”进入下一界面。

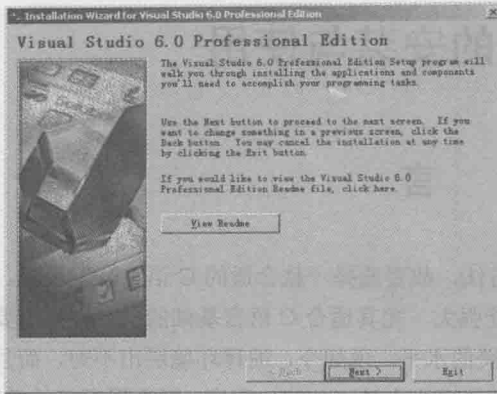


图 1-2 VS 6.0 安装图示 1

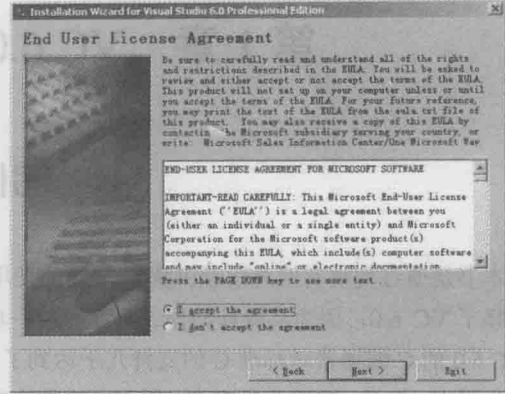


图 1-3 VS 6.0 安装图示 2

如图 1-4 所示, 此图为 ID 号以及姓名和单位的填写, ID 号可按图中填写, 姓名和单位由本人自有填写, 填写完成后, 用鼠标单击“Next”进入下一界面。

图 1-5 为呈现出的下一界面, 需要读者选择“Install Visual Studio 6.0 Professional”复选框进行下一步安装, 单击“Next”, 进入选择安装位置界面, 如图 1-6 所示。

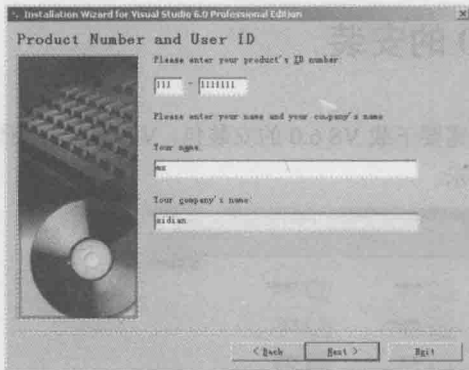


图 1-4 VS 6.0 安装图示 3

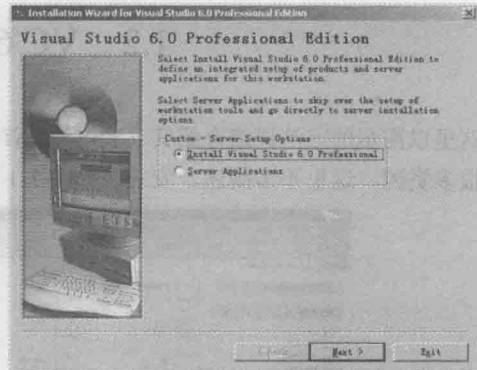


图 1-5 VS 6.0 安装图示 4

进入如图 1-6 所示界面, 将你所需的安装地址输入对话框, 或者单击“Browse”, 根据弹出的对话框选择安装目录。完成后单击“Next”进入下一步骤。

如图 1-7 和图 1-8 所示为之后两步, 这是两个系统提示, 只需单击“Continue”和“OK”便可继续安装程序。



图 1-6 VS 6.0 安装图示 5

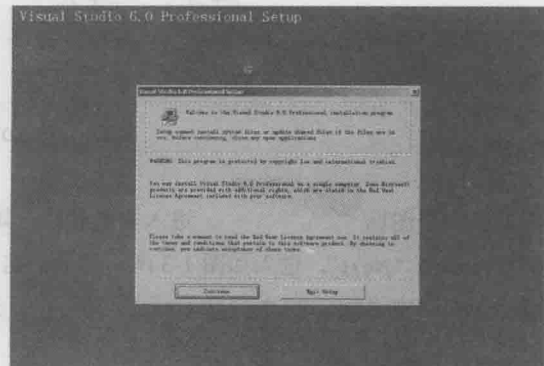


图 1-7 VS 6.0 安装图示 6

单击如图 1-8 所示的“OK”按钮后，便可进入如图 1-9 所示的安装界面，这里是选择安装方式，只有“Custom”一种，单击这个按钮。

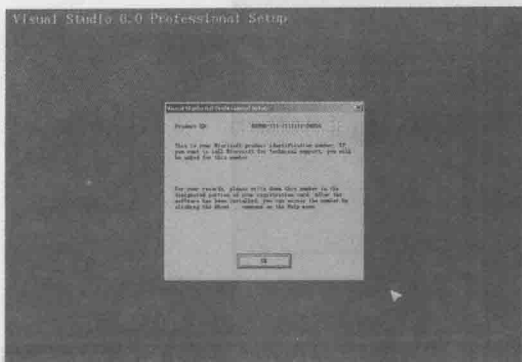


图 1-8 VS 6.0 安装图示 7

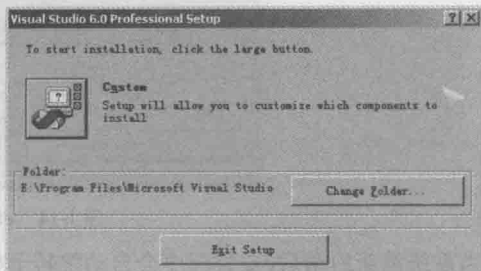


图 1-9 VS 6.0 安装图示 8

接下来进入如图 1-10 所示的界面，此界面是询问用户需要进行哪些组建的安装，这里选择默认状态，不做任何更改进入下一步操作。

单击图 1-10 中的“Continue”，进入如图 1-11 所示界面，此界面是询问是否需要注册环境变量，这里勾选它，选择注册环境变量。

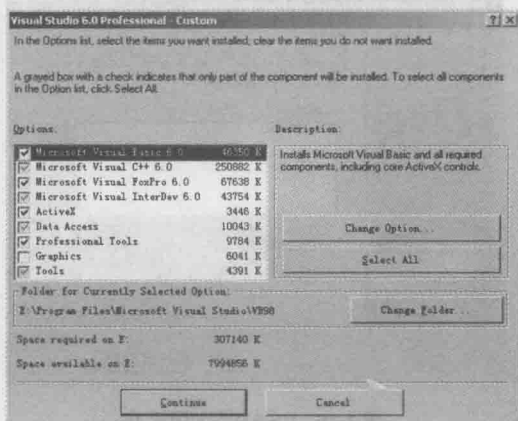


图 1-10 VS 6.0 安装图示 9

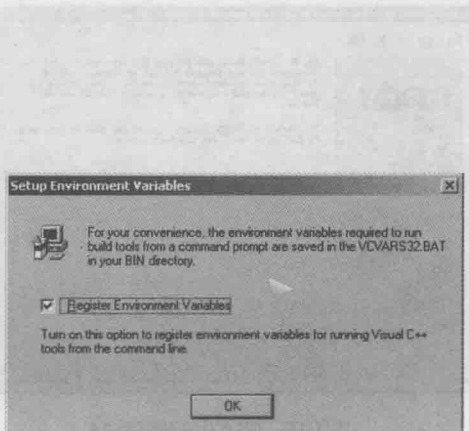


图 1-11 VS 6.0 安装图示 10

接下来便进入正式安装的过程，如图 1-12 所示。

当进度条走完会出现一个系统提示，如图 1-13 所示，单击“确定”便可。之后，会出现询问是否重新启动的对话框，如图 1-14 所示，这个过程是必须做的，在重启之后，系统会将环境变量等信息自动注册好，这样才能正常运行 VS 6.0，所以请保存好正在进行的工作，然后单击“Restart Windows”按钮。

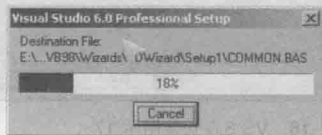


图 1-12 VS 6.0 安装图示 11

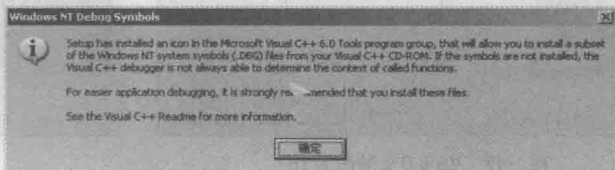


图 1-13 VS 6.0 安装图示 12

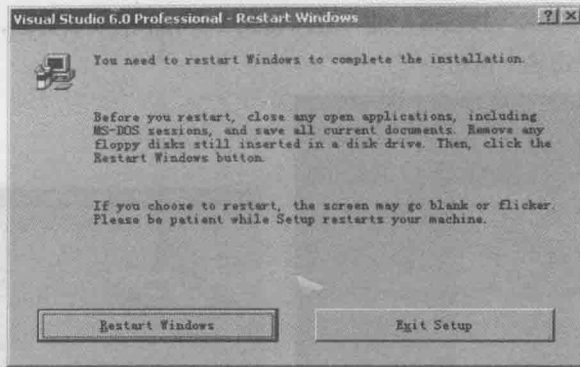


图 1-14 VS 6.0 安装图示 13

当系统重新启动后,安装程序会自动跳转至如图 1-15 的界面,此界面询问是否安装 MSDN。MSDN 是为方便程序员查询系统函数等官方提供的技术文档,如果读者需要使用可以下载 MSDN 的安装包,这里选择不安装。

接下来,安装程序还会提示两项安装以及在线注册,这些一般是用不到的,所以通常均选择不安装,如图 1-16,图 1-17,图 1-18 所示。

最后单击如图 1-18 所示中的“Finish”按钮,整个安装过程便顺利完成了。

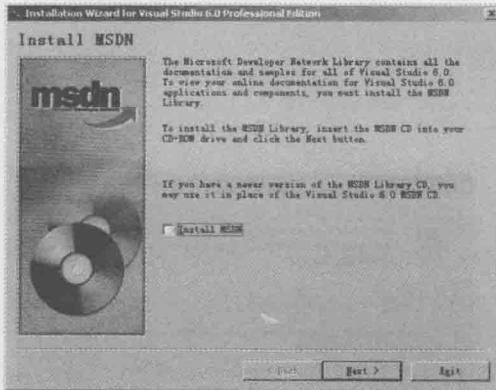


图 1-15 VS 6.0 安装图示 14

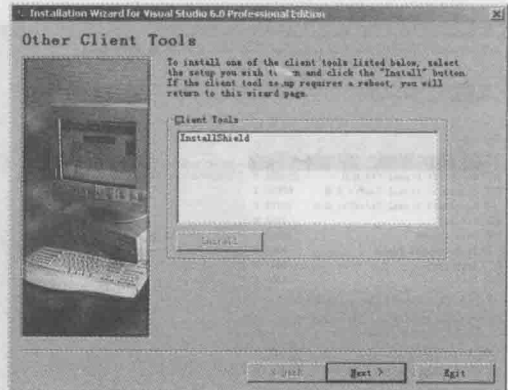


图 1-16 VS 6.0 安装图示 15

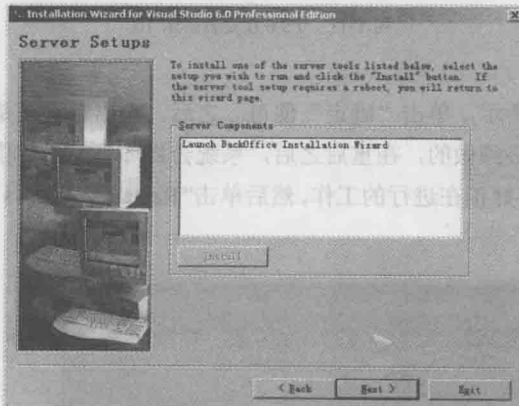


图 1-17 VS 6.0 安装图示 16

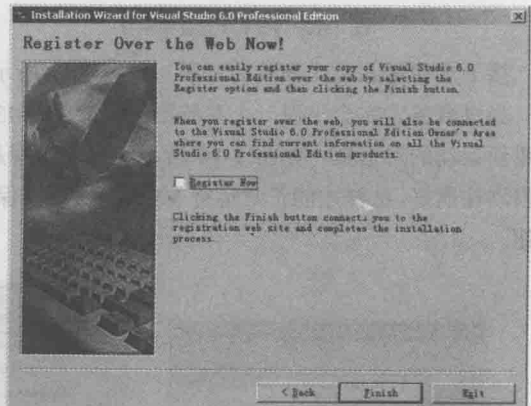


图 1-18 VS 6.0 安装图示 17

## 1.3 VC 6.0 的使用

VC 6.0 是一个非常强大的 C 编译环境,在这一节首先讲解一下这其中的基础操作,包括如何建立编译.c 文件,如何进行调试,这些都是应用 C 语言时所必须的,至于如何构建工程,将会在后面具体的章节中讲述。更为细致的工程,如构建 MFC 这样的窗口程序工程,就已经不再属于 C 语言基础范畴了,如果读者有兴趣,可以查阅针对其具体内容的书籍。

### 1.3.1 编译.c 文件

如果读者已经完成了 VC 6.0 的安装,那么从现在开始就正式进入编程阶段。首先从一个最简单的例子开始,这也基本上是所有程序员的第一个程序。

第一步,打开安装好的 VC 6.0,构建一个.c 文件。如果读者已经顺利地安装,那么单击开始菜单,就会提示安装了新程序,在所有程序中找到已安装好的 VS 6.0,如图 1-19,图 1-20 所示。接着,在 VS 6.0 中选择如图 1-21 所示的对应项,单击鼠标左键。这样就进入了 VC 6.0 编译环境,如图 1-22 所示。



图 1-19 构建.c 文件图示 1



图 1-20 构建.c 文件图示 2

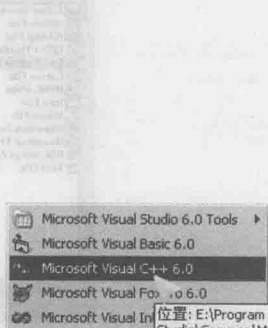


图 1-21 构建.c 文件图示 3

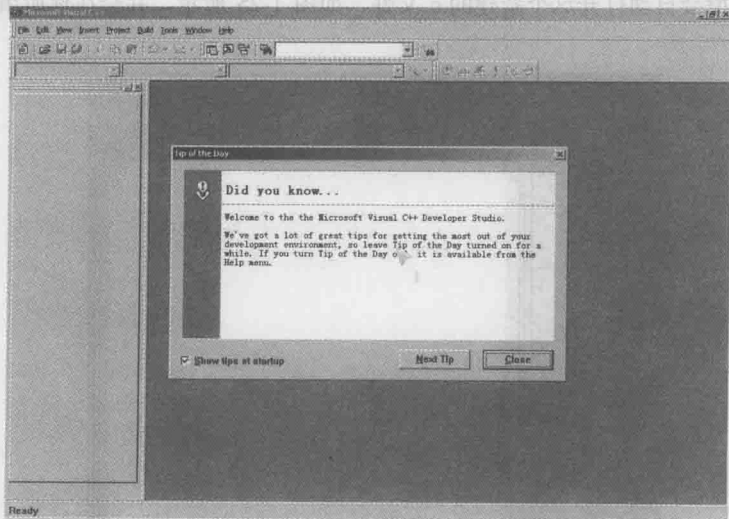


图 1-22 构建.c 文件图示 4

现在就来正式编写第一条程序，输出语句“hello world!”。新建一个.c 文件，单击工具栏的“File”，在弹出的目录中单击“New”，如图 1-23 所示。

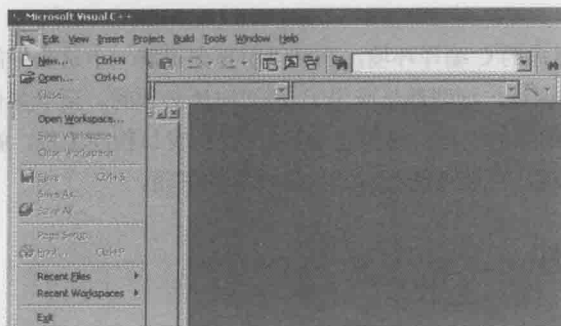


图 1-23 构建.c 文件图示 5

接着会弹出如图 1-24 所示的对话框，选择“Files”选项卡，选中其中的“C++ Source File”项，并在右侧的“File”框中输入“test.c”。底下的“Location”是存放该文件的目录，可以通过手动填写及选择路径的方式，将此文件添加到任意位置，所有工作完成之后单击“OK”按钮。

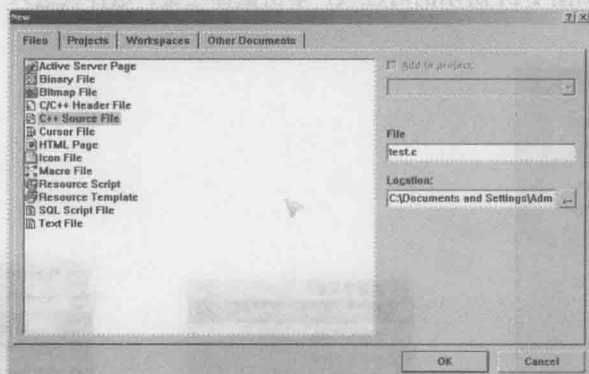


图 1-24 构建.c 文件图示 6

这时，编译环境会自动打开这个新建的.c 文件，如图 1-25 所示，并会在你所填写的位置生成一个相应的.c 文件。

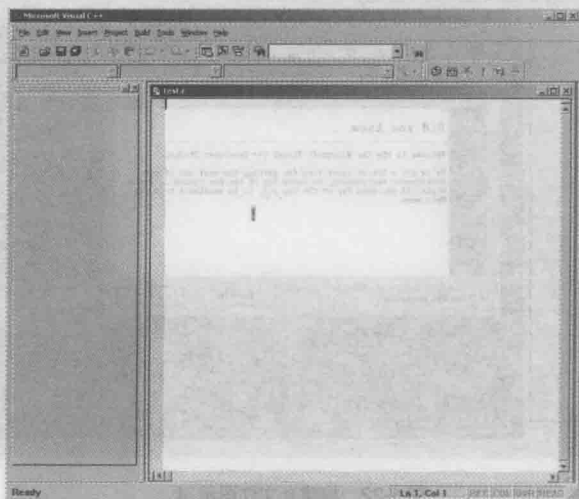


图 1-25 构建.c 文件图示 7

现在就可以开始编写代码了，在图 1-25 中 test.c 窗口就是书写代码的地方，将如下代码写入 test.c 中：

```
#include<stdio.h>
main()
{
    printf("hello world!\n");
}
```

接下来就要编译并运行这个文件，单击如图 1-26 所示的功能键，左侧的键是编译该文件，当前键是编译并建立链接，所以可以直接单击当前键进行编译和链接。



图 1-26 文件编译 1

单击后会在编译环境最下方出现一个对话框显示如图 1-27 所示的信息，这个对话框大家以后会经常用到，编译信息会在这里进行提示，包括报错、警告等。图示的信息表示一切正常。

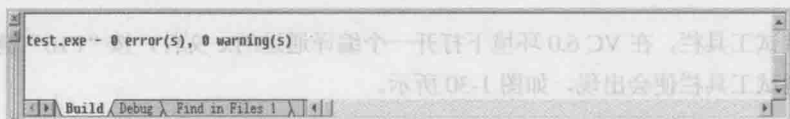


图 1-27 文件编译 2

接下来就可以运行这个文件了，在图 1-26 所示的功能键中，会发现原本灰色的感叹号变成了红色，这个按钮就是运行按钮，如图 1-28 所示。用鼠标单击它便会出现运行结果，如图 1-29 所示。



图 1-28 文件运行

到这里，第一个程序就顺利的完成了，同时也验证了编译环境是可以正常使用的。如果在编译或运行时出现了异常，就说明 VS 6.0 不能正常工作，这时就要检查是否正确安装，安装包是否存在问题，以及与操作系统及其他软件是否兼容。这里推荐使用 Windows XP 操作系统，更高版本的操作系统会出现不兼容的问题。



图 1-29 文件运行结果

## 1.3.2 调试

“调试”这个词会在编程者生涯当中经常听到，相信也会是最不愿意听到的词。C 语言虽然已经是高级语言，具有很强的逻辑性，但其表达能力还远不能与自然语言相提并论。所以当编写程序的时候，不可能像检查作文或者数学题那样通过通读全文来纠正错误。这个时候就要用调试工具来帮助订正逻辑错误，所以调试就是纠正程序错误，使其能够正常运行，得到正常结果的过程。当然，在修改逻辑错误之前首先应该保证语法正确，让程序能够运行起来，这也包含在调试范畴之内，这个过程主要是通过编译信息对话框的错误信息提示进行修改，在这里就不具体介绍了。与此同时，在读别人写的程序时，想直接通过源代码完全理解程序对于一般人来说往往是很难的，但通过调试工具的提示，会简化这个过程。下面介绍 VC 6.0 的调试方法。

### 1. 单步调试

程序调试的核心就是单步执行，所谓单步执行就是让程序一行一行的运行，再大的程序也是由每一行组成的，程序出错也都是到了特定行出现了错误，只有找到这些错误的源头，才能纠正程序的错误。

首先介绍调试工具栏，在 VC 6.0 环境下打开一个编译通过的.c 文件，按“F10”键，这时便进入了调试环境，调试工具栏便会浮现，如图 1-30 所示。

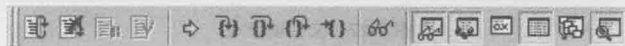


图 1-30 调试工具栏

单步执行主要用到的是图标为大括号的这几个按钮，第一个是跳入函数（“F11”），第二个是不跳入函数单步运行（“F10”），第三个是跳出函数（“Shift+F11”），第四个不太常用，是直接运行到光标所在行。因为要一步一步的运行程序，所以推荐大家使用括号内的快捷键进行操作。这里举例来解释如何使用快捷键。

在 VC 6.0 下新建一个.c 文件 test.c，在其中输入如下代码：

```
#include<stdio.h>
void A(int num,int* res)
{
    int i;
    for(i=0;i<10;i++)
    {
        res[i] += num;
    }
}
void B(int num,int* res)
{
    int i;
    for(i=0;i<10;i++)
    {
```