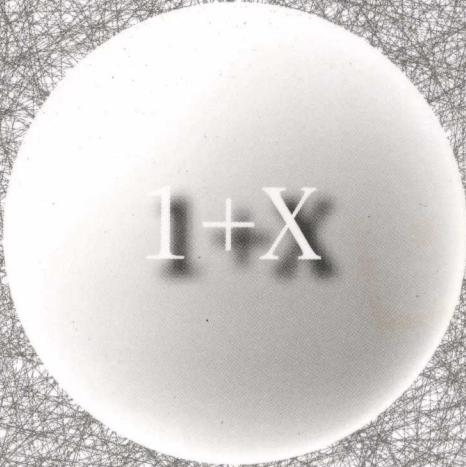


大学计算机基础教育规划教材

丛书主编 冯博琴

C程序设计导引

孙燮华 编著



1+X

清华大学出版社



九江学院图书馆



1816555

1522282

大学计算机基础教育规划教材

丛书主编 冯博琴

C程序设计导引

孙燮华 编著

TP312/21211

14X

不外借

九江学院图书馆
藏书章

清华大学出版社
北京

内 容 简 介

本书除了介绍 C 语言基础、基本数据类型、各种运算符与表达式、控制结构与语句、数组、函数、指针、结构体、文件等基本内容外,加强了算法的设计和编程能力的培养,还特别介绍 C 图形程序在 Visual C++ 6.0 环境中的编译和运行。本书注意培养学生的编程和创新能力,尤其注重可视化的图形编程和算法设计能力的培养,为后续学习面向对象编程和图形学与图像处理等课程作基础准备。为提高读者对学习 C 语言的兴趣,本书还提供了 C 语言应用于图形图像处理、计算机密码学、通信、数据压缩、智能算法和动画等领域的应用实例。

本书可作为高等学校理工科各专业 C 语言程序设计教材,也可作为计算机等级考试教学和自学用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 程序设计导引/孙燮华编著. —北京:清华大学出版社,2011.11

(大学计算机基础教育规划教材)

ISBN 978-7-302-26323-4

I. ①C… II. ①孙… 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(2011)第 152352 号

责任编辑:张 民 战晓雷

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185×260

印

张:23.75

字

数:562 千字

版 次:2011 年 11 月第 1 版

印

次:2011 年 11 月第 1 次印刷

印 数:1~4000

定 价:35.00 元

产品编号:042451-01

序

大学计算机基础教育规划教材



进入 21 世纪,社会信息化不断向纵深发展,各行各业的信息化进程不断加速。我国的高等教育也进入了一个新的历史发展时期,尤其是高校的计算机基础教育,正在步入更加科学,更加合理,更加符合 21 世纪高校人才培养目标的新阶段。

为了进一步推动高校计算机基础教育的发展,教育部高等学校计算机科学与技术教学指导委员会近期发布了《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求》(以下简称《教学基本要求》)。《教学基本要求》针对计算机基础教学的现状与发展,提出了计算机基础教学改革指导思想;按照分类、分层次组织教学的思路,《教学基本要求》提出了计算机基础课程教学内容的知识结构与课程设置。《教学基本要求》认为,计算机基础教学的典型核心课程包括大学计算机基础、计算机程序设计基础、计算机硬件技术基础(微机原理与接口、单片机原理与应用)、数据库技术及应用、多媒体技术及应用、计算机网络技术及应用。《教学基本要求》中介绍了上述六门核心课程的主要内容,这为今后的课程建设及教材编写提供了重要的依据。在下一步计算机课程规划工作中,建议各校采用“1+X”的方案,即“大学计算机基础”+ 若干必修或选修课程。

教材是实现教学要求的重要保证。为了更好地促进高校计算机基础教育的改革,我们组织了国内部分高校教师进行了深入的讨论和研究,根据《教学基本要求》中的相关课程教学基本要求组织编写了这套“大学计算机基础教育规划教材”。

本套教材的特点如下:

- (1) 体系完整,内容先进,符合大学非计算机专业学生的特点,注重应用,强调实践。
- (2) 教材的作者来自全国各个高校,都是教育部高等学校计算机基础课程教学指导委员会推荐的专家、教授和教学骨干。
- (3) 注重立体化教材的建设,除主教材外,还配有多媒体电子教案、习题与实验指导,以及教学网站和教学资源库等。
- (4) 注重案例教材和实验教材的建设,适应教师指导下的学生自主学习的教学模式。
- (5) 及时更新版本,力图反映计算机技术的新发展。

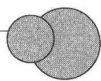
本套教材将随着高校计算机基础教育的发展不断调整,希望各位专家、教师和读者不吝提出宝贵的意见和建议,我们将根据大家的意见不断改进本套教材的组织、编写工作,为我国的计算机基础教育的教材建设和人才培养作出更大的贡献。

“大学计算机基础教育规划教材”丛书主编
教育部高等学校计算机基础课程教学指导委员会副主任委员

冯博琴



前言



C 语言的编译环境已从传统的 Turbo C 2.0 变为 Visual C++ 6.0。由于 Turbo C 2.0 中进行图形编程的 Graphics.h 与 Visual C++ 6.0 不相容,这一改变使大部分 C 语言教材没有图形编程一章,即使有 C 语言图形编程内容的教材,其编程环境仍然回到 Turbo C 2.0 中。这表明 C 语言编译环境不完全适应图形编程。另一方面,计算机的可视化,即图形化是当前的主流,Windows 窗体化已经在计算机领域取得了统治地位。由于这些原因,作者在本书中引入了属于 Windows 编程的 API,稍微向 Windows 跨出一步。对于不熟悉 Windows 编程的读者,只要输入本书的 Windows.c 程序,引用这个文件后,即可如通常编写 C 程序一样编写 C 图形图像程序了。

与目前出版的所有 C 语言教材相比,本书独具的三大特色是:

(1) 适应在 Visual C++ 6.0 环境中进行图形图像编程。本书引入了一个属于 Windows 编程的 Windows.c 程序,只要将这个程序输入或略作修改,就可以用通常的方法进行图形图像编程。

(2) 除专设第 10 章算法设计与分析外,有关算法和编程的内容贯穿全书,并结合高等数学的方法进行分析,以培养学生应用数学方法设计算法和编程的能力。

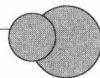
(3) 为扩大读者的知识面和提高学习 C 语言的兴趣,也为后续学习课程,如计算机图形学、图像处理、密码学、通信编码和游戏动画等作基础准备,本书提供了动态 Bezier 曲线、显示 PGM 图像、序列加/解密、通信 GCD 算法和动画小球碰撞等应用的实例。

本书在成书过程中得到了周永霞博士、姚伏天博士和吴永刚等老师的帮助,特此表示感谢。对本书引用的参考书的作者们表示感谢。

本书由王德林副教授撰写第 1~8 章,由孙燮华撰写第 9~12 章,全书由孙燮华定稿。由于作者的能力和水平有限,虽然经过了极大的努力,书中仍然难免还有一些错误,希望读者和同行专家批评指正。

作者

2011 年 5 月于杭州



第 1 章 概论	1
1.1 C 语言的发展与特点	1
1.1.1 C 语言发展简史	1
1.1.2 C 语言的特点	2
1.2 第一个 C 程序	3
1.2.1 第一个 C 程序及其编译与运行	3
1.2.2 关于 C 程序的编写	6
1.2.3 C 程序的结构	7
1.3 本书提供的 C 语言应用实例介绍	10
1.3.1 概述	10
1.3.2 本书部分应用举例	11
习题 1	13
第 2 章 C 语言基础	15
2.1 字符集、标识符与关键词	15
2.1.1 字符集	15
2.1.2 标识符	15
2.1.3 关键词	16
2.2 数据类型	16
2.2.1 基本数据类型	17
2.2.2 其他数据类型	18
2.3 常量与变量	18
2.3.1 常量	18
2.3.2 变量	22
2.4 运算符与表达式	25
2.4.1 赋值运算符与表达式	25
2.4.2 算术运算符与表达式	27
2.4.3 关系运算符与表达式	29
2.4.4 逻辑运算符与表达式	30

2.4.5	逗号运算符与表达式	32
2.4.6	复合运算与表达式	34
2.4.7	运算符优先级	34
2.5	数据输入/输出函数与常用函数	36
2.5.1	数据输入/输出函数	36
2.5.2	字符输入/输出函数	40
2.5.3	常用函数	41
习题 2		44
第 3 章	程序控制流与程序算法设计	47
3.1	顺序结构	47
3.1.1	顺序结构的基本语句	48
3.1.2	顺序结构应用举例	49
3.2	选择结构	52
3.2.1	if 语句	52
3.2.2	条件表达式	56
3.2.3	switch 语句	57
3.3	循环结构	59
3.3.1	while 循环	59
3.3.2	do-while 循环	62
3.3.3	for 循环	63
3.3.4	循环语句的嵌套	66
3.3.5	break 语句和 continue 语句	67
3.3.6	goto 语句	68
3.4	算法	68
3.4.1	算法概述	68
3.4.2	算法描述方法	71
3.5	程序设计方法	77
3.5.1	程序设计的一般步骤	77
3.5.2	结构化程序设计方法	77
习题 3		79
第 4 章	函数	83
4.1	函数声明与定义	83
4.1.1	概述	83
4.1.2	函数声明	86
4.1.3	函数定义	88
4.2	函数调用	89

4.2.1	传值调用的特点	89
4.2.2	函数调用方式	91
4.2.3	函数嵌套调用	92
4.2.4	函数递归和递推调用	94
4.3	函数与变量	97
4.3.1	全局变量和局部变量	98
4.3.2	变量的生命期与存储类别	100
习题 4		103
第 5 章	数组与字符串	108
5.1	一维数组	108
5.1.1	一维数组定义	108
5.1.2	一维数组引用	109
5.1.3	数组元素的排序与查找	111
5.2	二维数组	116
5.2.1	二维数组定义	116
5.2.2	二维数组引用	118
5.2.3	二维数组程序举例	120
5.3	字符串	120
5.3.1	字符串的存储	121
5.3.2	字符串的输入与输出	122
5.3.3	常用字符串函数	124
5.4	数组与函数	127
5.4.1	数组元素作为函数参数	127
5.4.2	数组名作为函数参数	127
习题 5		128
第 6 章	指针与动态内存分配	133
6.1	指针基础	133
6.1.1	指针变量定义	133
6.1.2	指针变量初始化	134
6.1.3	指针的间接引用	135
6.1.4	指针基本运算	136
6.2	指针与数组	140
6.2.1	指向数组元素的指针	140
6.2.2	通过指针引用数组元素	142
6.2.3	数组名作函数参数	143
6.2.4	指针数组	145

6.2.5	指向指针的指针	145
6.2.6	指针数组与二级指针的等价性	149
6.3	指针与字符串	150
6.3.1	字符串的表示形式	150
6.3.2	字符串指针作函数参数	151
6.3.3	常用字符串函数	152
6.4	指针与函数	153
6.4.1	指针作函数的参数	153
6.4.2	指针作函数返回值	157
6.4.3	指向函数的指针	158
6.4.4	函数指针作函数的参数	160
6.5	const 指针	161
6.5.1	指向常量的指针	161
6.5.2	指针常量	162
6.5.3	指向常量的指针常量	162
6.6	动态内存分配	163
6.6.1	用 malloc() 分配内存空间	163
6.6.2	动态数组的实现	165
习题 6		167
第 7 章	结构体、共用体和枚举类型	171
7.1	结构体类型	171
7.1.1	结构体类型及其变量的定义	171
7.1.2	结构体变量的使用	174
7.1.3	结构体变量的初始化	175
7.2	结构体数组	177
7.2.1	结构体数组的定义	177
7.2.2	结构体数组的初始化	177
7.3	结构体指针变量	178
7.3.1	指向结构体的指针	179
7.3.2	指向结构体数组的指针	180
7.3.3	结构体变量和结构体指针变量作函数参数	181
7.4	链表	183
7.4.1	链表概述	183
7.4.2	链表的基本操作	183
7.5	共用体	188
7.5.1	共用体类型的定义	188
7.5.2	共用体类型变量的定义	189

7.5.3 共用体变量的使用·····	189
7.6 枚举类型·····	190
习题7·····	192
第8章 输入/输出与文件 ·····	195
8.1 概述·····	195
8.1.1 C语言文件的概念·····	196
8.1.2 文本文件和二进制文件·····	197
8.1.3 文件类型指针·····	198
8.2 文件打开与关闭·····	198
8.2.1 打开文件·····	198
8.2.2 关闭文件·····	200
8.2.3 文件缓冲区的控制·····	201
8.2.4 文件状态检测·····	203
8.3 文本文件读/写·····	204
8.3.1 按字符读/写·····	204
8.3.2 按字符串读/写·····	206
8.3.3 格式化读/写·····	208
8.4 二进制文件读/写·····	210
8.4.1 写入二进制文件·····	210
8.4.2 读出二进制文件·····	212
8.5 文件定位与随机存取·····	213
8.5.1 文件定位·····	214
8.5.2 随机读/写·····	215
习题8·····	216
第9章 位运算及其应用 ·····	220
9.1 概述·····	220
9.1.1 位与字节·····	220
9.1.2 位序列的输出·····	223
9.2 位运算符及其使用方法·····	224
9.2.1 按位取反运算·····	224
9.2.2 按位左移运算·····	226
9.2.3 按位右移运算·····	228
9.2.4 按位与运算·····	229
9.2.5 按位或运算·····	232
9.2.6 按位异或运算·····	234
9.2.7 复合位运算符·····	237

9.3 位运算应用举例	237
9.3.1 位运算用于分离 IP 地址	237
9.3.2 均匀二分查找	239
9.3.3 二进制 GCD 算法	241
9.3.4 在计算机密码学中的应用	242
9.3.5 在数据压缩中的应用	246
习题 9	251
第 10 章 算法设计与分析	254
10.1 穷举法	254
10.1.1 百钱百鸡问题	254
10.1.2 提高穷举效率	255
10.1.3 局部穷举	260
10.2 分治法	263
10.2.1 二分法求解方程	263
10.2.2 快速排序法	265
10.3 回溯法	268
10.3.1 递归回溯法	269
10.3.2 非递归回溯法	273
10.4 贪心算法	274
10.4.1 旅行商问题	275
10.4.2 删数问题	277
10.5 再论递归与递推算法	279
10.5.1 递归和递推的效率	279
10.5.2 递归算法非递归化	282
10.6 大整数相加算法	284
10.6.1 大整数相加算法设计与实现	284
10.6.2 大 Fibonacci 数的计算	289
习题 10	291
第 11 章 图形与图像编程	296
11.1 概述	296
11.1.1 Win32 API 简介	296
11.1.2 Windows 应用程序的数据类型	297
11.2 Windows 应用程序结构	298
11.2.1 WinMain 函数	298
11.2.2 WndProc 窗口函数	302
11.3 图形编程	303

11.3.1 创建图形编程空工程	303
11.3.2 编写窗口程序	304
11.3.3 编写图形程序	306
11.3.4 常用图形函数介绍	306
11.3.5 图形编程实例	308
11.3.6 图形动画	317
11.4 图像编程	320
11.4.1 RAW 图像数据及其显示	320
11.4.2 PGM 和 PPM 图像显示和存储	324
习题 11	331
第 12 章 编译预处理与程序调试	334
12.1 编译预处理概述	334
12.1.1 预处理指令	334
12.1.2 文件包含	335
12.2 宏定义与宏替换	336
12.2.1 对象式宏	336
12.2.2 函数式宏	336
12.3 条件编译	338
12.3.1 条件指令 #if、#else、#elif、#endif 的使用	338
12.3.2 条件指令 #ifdef、#ifndef 的使用	340
12.4 调试语法错误	341
12.4.1 常见语法错误与警告的英文解释	342
12.4.2 常见语法错误	342
12.5 调试连接错误	348
12.6 调试逻辑算法错误	349
12.6.1 调试方法	349
12.6.2 常见逻辑错误	353
12.6.3 调试方法总结	355
12.7 运行错误	356
习题 12	357
附录 A ASCII 字符代码	359
附录 B C 语言常用库函数	360
参考文献	364

第1章

概 论



我们已经学习了“大学计算机基础”或者“大学计算机文化”课程,能使用计算机上网浏览网页,或在计算机中输入文字等。所有这些都是人指挥计算机进行的工作。计算机之所以能进行某项工作,是因为这项工作已由程序员预先设计了需要实现的指令,计算机才能够按照预先设计的指令有条不紊地工作。这些指令就组成了程序。若没有程序,计算机将“不知所措”,无法运行。编写指令需要使用能与计算机沟通的“语言”。早期相当多的软件是用 C 语言写成的。目前,C 已经发展成 C++、C# 等系列语言,Microsoft 公司还开发了将 C++ 和 C# 等可视化的 Visual C++、Visual C++ .NET 和 Visual C# .NET 等集成开发环境。虽然 C++、C# 的功能更为强大,但其基础仍然是 C 语言。C 语言是计算机的主要基础语言。如同微积分学是整个现代数学的主要基础一样。

本书将紧密地结合实例介绍编程的方法和技巧,并特别注重培养编程与算法设计和分析的能力。

1.1 C 语言的发展与特点

在众多的程序设计语言中,C 语言有其独特之处,深受软件工作者欢迎。本节主要从程序设计的角度介绍 C 语言的发展及其特点。

1.1.1 C 语言发展简史

在 C 语言诞生以前,系统软件主要是用汇编语言写成的。由于汇编语言程序依赖于计算机硬件,所以其可读性和可移植性都很差;而一般的高级语言又难以实现对计算机硬件的直接操作,于是人们盼望能有一种兼有汇编语言和高级语言特性的新语言。

C 语言就是在这种背景下于 20 世纪 70 年代初问世的。当时 C 语言主要用于 UNIX 系统的开发。UNIX 操作系统最初是在 1969 年由美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 用汇编语言开发而成的。后改用 B 语言实现,但 B 语言过于简单,功能有限。为了更好地描述和实现 UNIX 操作系统,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最初的 C 语言。几经修改后,于 1978 年正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合写了著名的“The C Programming Language”一书,通常简称为“K&R”,也有人称之为“K&R 标准”。但是,在“K&R”中并

没有定义一个完整的标准 C 语言,后来由美国国家标准化协会 ANSI(American National Standards Institute)在此基础上制定了 C 语言标准,于 1983 年发表,通常称之为 ANSI C。

随着人们对 C 语言的强大功能和各方面优点的逐步了解,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之一。

近年来,由于开发大型软件的需要,C++ 在我国逐步得到推广。C++ 是面向对象的程序设计语言,其基础是 C 语言,且二者在很多方面是兼容的。学习 C++ 要比 C 语言困难些,并且也不是所有的人都去编写大型软件。因此,在国内外的大学中,都把 C 语言作为一门重要的课程。掌握了 C 语言,再去学习 C++,就会达到事半功倍的效果。

因为 C++ 是由 C 语言发展而来的,C++ 将 C 作为一个子集包含了进来。所以,使用 C 语言编写的程序可以用 C++ 编译系统进行编译。C/C++ 的编译系统有很多,如 GCC(GNU Compiler Collection)家族的 Dev-C++ (Mingw32)、Cygwin, Borland 公司的 Turbo C 2.0(只支持 C 语言)、Borland C++,微软公司的 Visual C++,以及 LCC-Win32 等。它们不仅实现了 ANSI C 标准,而且还各自做了一些扩充,使之更加方便和完美。但是,它们在实现 C 语言时略有差异,请读者参阅相应的手册,并注意自己在上机时所使用的 C 编译系统的特点和规定。

本书叙述以 ANSI C 为主,上机实验使用 Visual C++ 6.0(以后简称 VC 6.0 或直接简称 VC),与全国计算机等级考试的 C 语言环境一致。

1.1.2 C 语言的特点

1. 与自然语言比较

自然语言是人类交流的重要工具,人类可以利用自然语言进行沟通和交流,共同完成生产和生活实践。人与计算机进行交流也是属于信息的交流,只是表达方式、规则等与人类交流有所不同。自然语言与 C 语言的比较如表 1-1 所示。

表 1-1 自然语言与 C 语言比较

异同点	自然语言	C 语言
信息交流	交流双方地位平等,均有思维和推理能力	采取命令方式人机对话,计算机一般无思维和推理能力,但具有计算与逻辑判断能力
语法规则与句法规则	规则灵活,可省略、颠倒部分内容。如: “走,上课去!” “上课去,走!” (上面两句话意思相同,省略主语)	规则固定,一般不可省略、颠倒,必须按规则。例如: <code>scanf("%d%d",&a,&b);</code> <code>x=a+b;</code> (若两条语句颠倒可能得不到正确结果)
表达方式	多样	算法多样

2. 与其他程序设计语言比较

与其他程序设计语言相比较,C 语言有如下的特点。

(1) C语言简洁、紧凑,使用方便、灵活。程序书写自由,主要用小写字母表示,压缩了一切不必要的成分。

(2) 运算符丰富。C语言把括号、赋值、逗号等都作为运算符处理,从而使C语言的运算类型极为丰富,可以方便地实现其他高级语言难以实现的功能。

(3) 数据结构类型丰富,具有现代语言的各种数据结构。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。能实现各种复杂数据结构(如链表等)的运算。尤其是指针类型数据,使用起来更为灵活、多样。

(4) 具有结构化的控制语句。用函数作为程序的基本单位,便于实现程序的模块化。C语言是良好的结构化语言,符合现代编程风格的要求。

(5) 语法限制不太严格,程序设计自由度大。如对数组下标越界不做检查;对变量的类型使用比较灵活,如整型数据与字符型数据可以通用。

(6) C语言允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此有人把它称为中级语言。

(7) 生成的目标代码质量高,程序执行效率高,可达到汇编语言程序约80%的功能。

(8) 与汇编语言相比,用C语言写的程序可移植性好。

C语言是理想的结构化语言,描述能力强,且现在的操作系统课程大多结合UNIX讲解,而UNIX与C不可分。因此,C语言已经成为广泛使用的教学语言。C除了能用于教学外,还有广阔的应用领域,因此更有生命力。Pascal和其他高级语言的设计目标是通过严格的语法定义和检查来保证程序的正确性,而C语言则是强调灵活性,使程序设计人员能有较大的自由度,以适应宽广的应用面。“限制”与“灵活”是一对矛盾。限制严格,就失去灵活性;而强调灵活,就必然增加了出错的可能性。一个不熟练的程序设计人员编写一个正确的C程序可能会比编写一个其他高级语言程序更难一些。也就是说,对使用C语言的人,要求对程序设计更熟练一些。总之,C语言对程序员要求较高,但程序员使用C语言编写程序会感到限制少,灵活性大,功能强,可以编写出任何类型的程序。现在,C语言已不仅用来编写系统软件,也用来编写应用软件,因此C语言是在高等学校计算机课程教学中最普及的语言。

1.2 第一个C程序

用高级语言编写的程序称为**源程序**或**源代码**。计算机只能识别和执行由0和1组成的二进制指令,不能识别和执行用高级语言写的指令。为了使计算机能执行高级语言源程序,必须先用编译程序把源程序翻译成二进制形式的目标程序,然后将目标程序与系统的函数库和其他目标程序连接起来,形成可执行的目标程序。其中,源程序文件的后缀是.c,可执行文件的后缀是.exe,库函数文件的后缀是.h。

下面将介绍第一个C程序的编写、编译与运行。

1.2.1 第一个C程序及其编译与运行

目前,全国计算机等级考试大纲把C语言编译环境由Turbo C 2.0转变为Visual

C++ 6.0。计算机专业的多数初学者也把 Visual C++ 6.0 作为编译环境,这是因为 Visual C++ 6.0 在 Windows 下运行更易于上手操作。本书完全适应这个潮流,将 Visual C++ 6.0 作为 C 语言的编程工具。下面介绍 Visual C++ 6.0 中有关 C 语言的部分功能。

1. 启动 Visual C++ 6.0

双击 Visual C++ 6.0 启动图标^{2.2.1},其界面如图 1-1 所示。

2. 新建文件

在图 1-1 的菜单中,选择 File(文件)|New(新建),将出现“New(新建)”对话框,如图 1-2 所示。

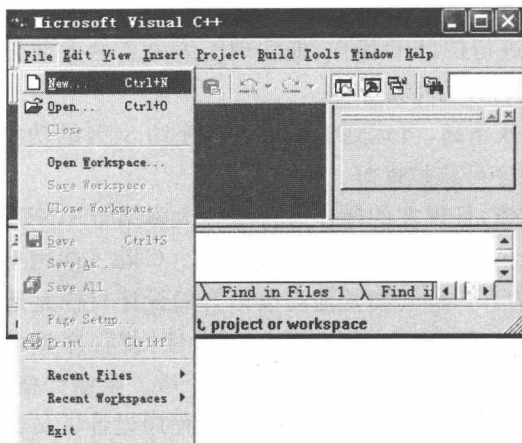


图 1-1 Visual C++ 6.0 界面

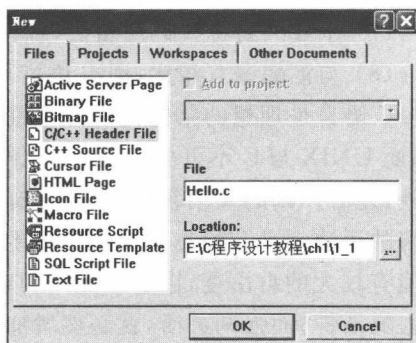


图 1-2 New(新建)对话框

选中对话框上部的 Files 选项卡,在左边一栏中,选中 C/C++ Header File 或 Text File;在右边 File 下面的文本框中填写程序文件名,例如,Hello.c,这里.c 表示 C 语言源程序,是程序名的关键部分,不能省略。单击 Location: 文本框右侧的图标^{2.2.2},在打开的文件目录中,选择 C 程序的存储目录,单击 OK 按钮,又回到第一次出现的界面。出现变化的是左上方区域已变成白色,如图 1-3 所示。

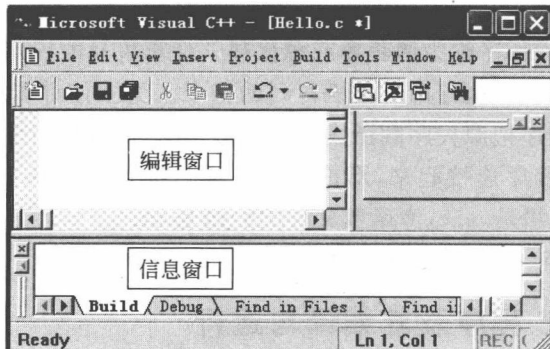


图 1-3 编辑窗口和信息窗口