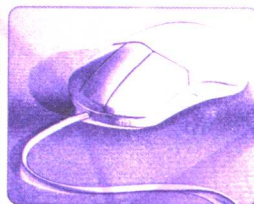
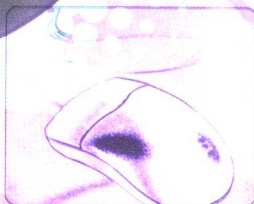
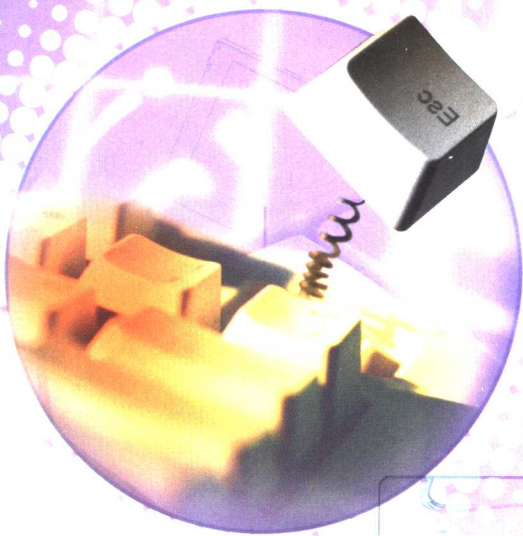


新 IT 人职业技能培训规划教材



李刚 等编著

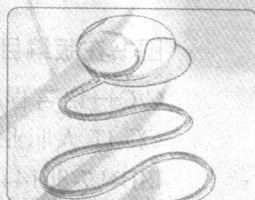
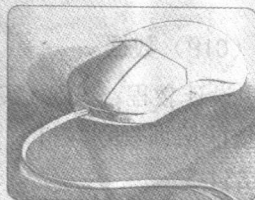
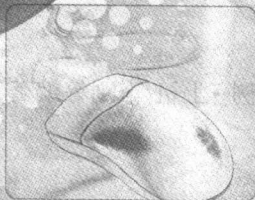
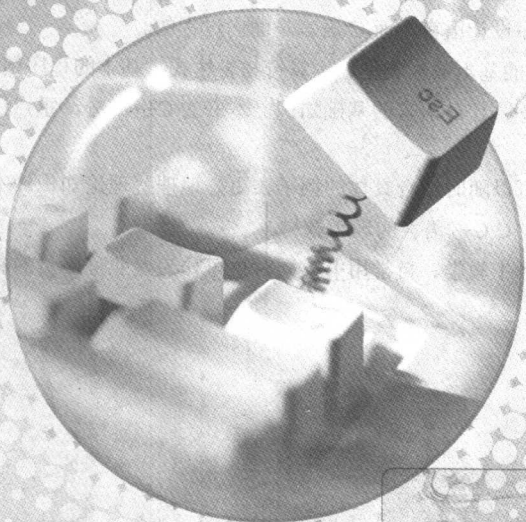
C/C++语言程序设计教程

——基础·实例·上机



中国宇航出版社

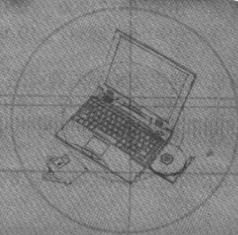
新 IT 人职业技能培训规划教材



李刚 等编著

C/C++语言程序设计教程

——基础·实例·上机



中国宇航出版社

内 容 简 介

C 语言是一种重要的计算机程序设计语言，它是为计算机程序员准备的专业程序设计语言。C++语言是在 C 语言基础上发展而来的一种面向对象的计算机程序设计语言，而 C 语言是面向过程的。

本书系统讲述了 C 语言的基础知识，包括 C 语言的基本数据类型、运算符与表达式、基本的程序结构、数组、函数、指针、位运算和文件等知识。这些内容是学习 C 语言的基础知识。本书的 C++部分，重点讲述了 C++面向对象的特性，这是 C++语言最重要的特性。

本书在讲述理论的同时配合大量的实例，让读者可以随时结合具体的例子去更好地理解理论知识。本书内容具有通俗易懂、理论性、实际性强等特点，是学习 C/C++语言的理想教材。

本书既可作为中等和高等职业技术院校相关专业师生教学、自学用书，同时也可作为专业程序员的参考书及社会 C 语言培训班的培训教程。

图书在版编目 (CIP) 数据

C/C++语言程序设计教程：基础·实例·上机/李刚等编著。—北京：中国宇航出版社，2004.9
(新 IT 人职业技能培训规划教材)

ISBN 7-80144-854-5

I. C... II. 李... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 083135 号

责任编辑：王 冬 黄鹏辉

审 校：李之聪

责任校对：刘冬艳

排 版：宇航计算机图书排版中心

出 版

发 行

社 址 北京市阜成路 8 号

邮 编 100830

经 销 新华书店

发行部 (010) 68371900 (010) 88530478 (传真)
(010) 68768541 (010) 68767294 (传真)

计算机 北京市和平里滨河路 1 号航天信息大厦 4 层
事业部 (010) 68372647 (010) 68372639 (传真)

邮 编 100013

承 印 北京云浩印刷有限责任公司

版 次 2004 年 9 月第 1 版

2004 年 9 月第 1 次印刷

规 格 787×1092

开 本 1/16

印 张 15.75

字 数 373 千字

印 数 1~5000 册

书 号 ISBN 7-80144-854-5

定 价 22.00 元

本书如有印装质量问题可与发行部调换

新 IT 人职业技能培训规划教材

编 委 会

主 任 邓宁丰

副主任 张铁钧

委 员 (按姓氏笔画排序)

张 鹏 王 冬 王小妮 宁勇峰 许哲平

齐洪喜 张素卿 李 刚 李之聪 邵 静

范 荣 宫鸣宇 徐楠楠 高 华 黄鹏辉

策 划 DGMOOK

丛书序言

我国人才需求调查表明, 计算机应用人才紧缺, 其中计算机网络、软件开发和多媒体开发领域的技术人才尤为短缺。虽然大中专院校每年都培养了大量的计算机应用人才, 但是仍然不能满足社会的实际需求。为此, 许多大中专院校都把计算机职业技能培训作为学校教学工作的重点。对于学校而言, 培养社会需要的计算机人才, 不但要有优良的办学设施和高水平的师资队伍, 高质量的教材也是必不可少的。然而, 从对大中专院校所使用计算机教材的调查情况看, 结果并不乐观。主要表现在教材内容传统, 学用脱节, 未侧重对学生操作技能和解决实际问题能力的培养。为此, 我们在研究现行同类教材的基础上, 结合国家对职业教育和职业培训的相关规定, 组织了一批具有丰富教学和教材编写经验的专家、学者, 精心编写了“新 IT 人职业技能培训规划教材”, 力求满足职业技术学院对计算机教材的需求。

“新 IT 人职业技能培训规划教材”是一套适合中等职业技术学校(包括职业高中和技校)、高等职业技术学院(包括高等专科学校)计算机或相关专业, 以及社会培训班使用的教材。本套教材具有以下特征。

一、学历证书与职业资格证书的“直通车”

教材内容涵盖国家相关职业标准(中级)的知识和技能要求, 通过系统学习, 学生可以达到通过职业资格认证的水平。

二、突出职业教育特色

从学生未来职业定位和人才市场的需求出发, 确定教材的体例和结构, 保证教材的实用性, 突出职业教育特色, 强调学生实际操作能力的培养。

三、保证教材编写质量

由具有一线培训和教学经验的教师执笔编写, 提供贴近教学实践的多种教学方案, 方便学校和教师选择与使用。

四、注重学生能力的培养

介绍最实用的计算机技术和最流行的软件, 结合大量实例, 强化学生的操作技能, 确保学生掌握的知识和技能在未来工作中应用自如。

五、语言简洁, 学习轻松

摒弃冗长、繁琐的理论性描述, 具有简洁、轻松的叙述风格, 详略得当, 篇幅适中, 在同类教材中是一套特点鲜明的简明实用教程。

首批出版的教材以文字处理与办公应用模块和程序编写模块为主, 计划陆续出版的教材以多媒体技术应用模块和网络技术应用模块为主, 详见下表。

序号	书 名	出版状态
1	计算机文化基础	√
2	汉字录入与办公软件应用	√
3	C/C++ 语言程序设计教程——基础·实例·上机	√
4	Visual Basic 程序设计教程——基础·实例·上机	√
5	Visual C++ 程序设计教程——基础·实例·上机	√
6	Visual FoxPro 数据库设计教程——基础·实例·上机	√
7	Delphi 程序设计教程——基础·实例·上机	√
8	计算机平面设计教程	*
9	计算机动画制作教程	*
10	计算机网页设计教程	*
11	计算机工程制图教程	*
12	计算机网络技术基础	*
13	计算机组网技术	*
14	计算机网络管理与安全	*
15	计算机组装与维修	*

注：√——首批出版；*——后续出版。

职业和技术教育是开启人类文明的钥匙，是现代社会进步的引擎，是保持科学和经济快速发展的重要条件。希望本套教材的出版能够对推动我国计算机职业教育贡献一份微薄之力，也希望读者将本套教材使用中的意见和建议及时反馈给我们。

最后，真诚感谢为本套教材编写、出版辛勤工作的所有朋友们，也祝愿读者能够早日完成学业，为社会发展作出更大贡献！

新 IT 人职业技能培训规划教材编委会

前 言

C 语言是现代程序设计中最活跃的高级语言之一。目前最流行的 Windows, Unix, Linux 三大操作系统都是用 C 语言编写的。虽然 C 语言和 C++ 语言比较难掌握, 但是它的功能却是最强大的。

本书内容涵盖了 C 语言的重要知识点, 但并不是将一个个知识点罗列出来, 而是本着先易后难、平滑过度、循序渐进、易读易懂的思路进行了整体结构设计。在结构体系方面充分注意到章节知识点的渐进性, 前后表述的一致性和疑难点的可释性。同时每一个知识点都配合一个或多个实例, 让读者能够更深入透彻的理解知识点。

另外, 本书的每一章后面大都附有上机练习和习题。上机练习是学习程序设计必不可少的部分。读者应该将书中的例题亲自上机运行一遍, 并按照自己的想法去更改程序的流程, 分析因此出现的结果。习题是巩固基础知识的有效方法, 本书中每一章的习题都是经过精心挑选的, 重在帮助读者巩固本章的基础知识。

本书共分 14 章, 第 1~2 章讲述了 C 语言程序设计的基础知识, 是最基础的部分; 第 3~5 章讲述了 C 语言程序设计的 3 种基本结构, 这是 C 语言程序的基本框架; 第 6~12 章讲述了 C 语言的几种比较复杂的数据类型, 以及比较高级的功能; 第 13~14 章讲述的是 C++ 的知识, 着重讲述了 C++ 面向对象的特性。教学内容和课时分配建议如下。

序号	教学内容	课时分配		
		授课	上机实践	合计
第 1 章	C 语言概述	1	0	1
第 2 章	数据类型、运算符与表达式	2	2	4
第 3 章	顺序结构程序设计	2	2	4
第 4 章	选择结构程序设计	2	2	4
第 5 章	循环结构程序设计	2	2	4
第 6 章	数组	3	2	5
第 7 章	函数	3	2	5
第 8 章	编译预处理 *	2	2	4
第 9 章	枚举与位运算	2	2	4
第 10 章	指针 *	3	2	5
第 11 章	结构体与共用体	3	2	5
第 12 章	文件	2	2	4
第 13 章	C++ 概述	2	2	4
第 14 章	C++ 面向对象基础	3	2	5
合计		32	26	58

本书适用于多层次教学和多目标培养, 对于学时数较少的专业可以仅学习不带星号(*)的章节, 对于学时数较多的专业除学习全部内容外, 应适当增加上机实践的时间。

在本书编写过程中, 徐日强、张素卿、于丙超、杜江、刘旭、郑砚等参加了部分素材整理, 李志伟、李子婷、陈策、李龙、吴金华、管佩森、张海霞、许社村、张俊岭等进行了部分内容的编写及审校工作, 在此一并表示感谢。

限于时间、水平和能力, 书中不妥之处, 恳请读者多加指教。

编 者

目 录

第 1 章 C 语言概述 1	4.2 switch 语句..... 35
1.1 C 语言的历史与现状..... 1	4.3 本章小结..... 38
1.2 C 语言的程序结构..... 1	4.4 上机实验..... 39
1.3 C 语言的开发环境..... 3	4.5 思考练习题..... 41
1.4 本章小结..... 5	第 5 章 循环结构程序设计 44
1.5 思考练习题..... 5	5.1 goto 语句..... 44
第 2 章 数据类型、运算符与表达式 6	5.2 while 语句..... 47
2.1 输入输出函数..... 6	5.3 do-while 语句..... 49
2.1.1 格式化输出函数 printf ()..... 6	5.4 for 语句..... 51
2.1.2 格式化输入函数 scanf ()..... 7	5.5 循环的嵌套..... 54
2.1.3 putchar () 和 getchar () 函数..... 8	5.6 几种循环的比较..... 55
2.2 C 语言的数据类型..... 9	5.7 break 和 continue 语句..... 55
2.2.1 常量和变量..... 9	5.7.1 break 语句..... 55
2.2.2 整型数据..... 10	5.7.2 continue 语句..... 56
2.2.3 字符型数据..... 11	5.8 本章小结..... 57
2.2.4 实型数据..... 12	5.9 上机实验..... 57
2.2.5 各类数值型数据之间的 混合运算..... 14	5.10 思考练习题..... 59
2.3 C 语言的运算符与表达式..... 15	第 6 章 数组 61
2.3.1 C 运算符简介..... 15	6.1 一维数组..... 62
2.3.2 算术表达式和运算符的 优先级和结合性..... 15	6.2 二维数组..... 66
2.3.3 赋值运算符和赋值表达式..... 17	6.2.1 二维数组元素的表示方法..... 66
2.3.4 逗号运算符和逗号表达式..... 18	6.2.2 二维数组的初始化..... 67
2.4 算法与流程图..... 19	6.3 字符数组..... 68
2.4.1 算法..... 19	6.4 字符串常用函数..... 71
2.4.2 算法的表示方法..... 21	6.5 本章小结..... 75
2.5 本章小结..... 22	6.6 上机实验..... 75
2.6 思考练习题..... 22	6.7 思考练习题..... 79
第 3 章 顺序结构程序设计 23	第 7 章 函数 81
3.1 结构化程序设计..... 23	7.1 函数的基本知识..... 81
3.2 C 语言的语句..... 24	7.1.1 函数的定义..... 81
3.3 本章小结..... 26	7.1.2 函数的参数和函数值..... 83
3.4 上机实验..... 27	7.2 函数的调用..... 85
3.5 思考练习题..... 29	7.2.1 函数调用的一般形式..... 85
第 4 章 选择结构程序设计 31	7.2.2 函数嵌套调用..... 87
4.1 if 语句..... 31	7.2.3 函数递归调用..... 92
	7.3 局部变量与全局变量..... 94
	7.3.1 局部变量..... 94
	7.3.2 全局变量..... 95

7.4	变量的存储类型	97
7.4.1	自动变量	98
7.4.2	外部变量	100
7.4.3	静态变量	101
7.4.4	寄存器变量	103
7.5	内部函数和外部函数	104
7.5.1	内部函数	104
7.5.2	外部函数	104
7.6	本章小结	105
7.6.1	C 语言中函数的分类	105
7.6.2	其他问题	105
7.7	上机实验	106
7.8	思考练习题	108
第 8 章	编译预处理	110
8.1	宏定义与符号常量	110
8.1.1	无参宏定义	110
8.1.2	有参宏定义	111
8.1.3	符号常量	113
8.2	文件包含和条件编译	113
8.2.1	文件包含	113
8.2.2	条件编译	114
8.3	本章小结	116
8.4	上机实验	116
8.5	思考练习题	117
第 9 章	枚举与位运算	119
9.1	枚举	119
9.1.1	枚举类型的定义和枚举 变量的说明	119
9.1.2	枚举类型变量的赋值和 使用	120
9.2	位运算	122
9.3	位域	126
9.4	本章小结	128
9.5	思考练习题	128
第 10 章	指针	130
10.1	指针和指针变量的概念	130
10.1.1	内存地址、变量地址和 变量值的存取	130
10.1.2	指针与指针变量	132
10.2	指针变量作函数参数	135

10.3	数组的指针和指向数组元素的 指针变量	137
10.3.1	数组的引用方式	137
10.3.2	指向多维数组的指针	140
10.4	字符串的指针和指向字符串的 指针变量	142
10.5	指针数组	145
10.6	返回指针值的函数	146
10.7	main 函数的参数	147
10.8	本章小结	148
10.9	上机实验	148
10.10	思考练习题	152
第 11 章	结构体与共用体	156
11.1	结构体类型变量的定义和引用	156
11.1.1	结构体类型变量的定义	157
11.1.2	结构体类型变量的引用	159
11.1.3	结构体类型变量的 初始化	160
11.2	结构体数组	161
11.3	指向结构体类型数据的指针	163
11.3.1	指向结构体类型变量的 指针	163
11.3.2	指向结构体类型数组的 指针	164
11.4	链表	166
11.4.1	单链表的定义与相关 操作	166
11.4.2	单链表的插入与删除	169
11.5	共用体	170
11.6	本章小结	173
11.7	上机实验	173
11.8	思考练习题	176
第 12 章	文件	177
12.1	文件的基本操作	177
12.1.1	文件的打开与关闭	177
12.1.2	文件的读写	181
12.1.3	文件检测函数	187
12.2	本章小结	188
12.3	上机实验	188
12.4	思考练习题	189

第 13 章 C++概述	191	13.7 思考练习题.....	201
13.1 面向对象的基本概念.....	191	第 14 章 C++面向对象基础	203
13.1.1 对象.....	191	14.1 类与对象.....	204
13.1.2 类.....	191	14.1.1 类的定义与对象的引用.....	204
13.1.3 消息.....	192	14.1.2 类与对象.....	206
13.1.4 方法.....	192	14.1.3 构造函数与析构函数.....	207
13.1.5 继承.....	192	14.1.4 函数重载.....	211
13.2 面向对象编程方法.....	192	14.1.5 友元.....	212
13.3 C++的开发环境.....	193	14.2 对象指针.....	216
13.3.1 中文版 Visual C++ 6.0 集成 开发环境.....	193	14.2.1 new 运算符和 delete 运算符.....	216
13.3.2 利用中文版 Visual C++ 6.0 集成开发环境创建第一个 C++程序.....	194	14.2.2 动态创建类对象.....	216
13.4 C++的输入输出.....	195	14.3 派生类与继承类.....	217
13.4.1 cout 流.....	196	14.3.1 单继承的派生类.....	217
13.4.2 cin 流.....	197	14.3.2 多继承的派生类.....	218
13.5 本章小结.....	198	14.4 本章小结.....	218
13.6 上机实验.....	199	14.5 上机实验.....	219
		14.6 思考练习题.....	220
		附录 A 参考答案	223

第 1 章 C 语言概述

本章要点

- C 语言的历史与现状。
- C 语言的程序结构。
- C 语言的常用开发环境。

这一章讲述了 C 语言的发展历史，并概括的讲述了 C 语言的一般格式以及最常用的 C 语言的开发环境。由于 C 语言的许多思想和设计理念是有其特殊的历史背景的，所以通过对本章的学习可以使读者对 C 语言有一个基本的了解，为以后各章的学习打下基础。

1.1 C 语言的历史与现状

C 语言是一种优秀的系统描述语言，它兼具低级语言的强大功能（例如，汇编语言可以直接对硬件操作）和高级语言良好的可读性及良好的可移植性等特点。它由早期的编程语言 BCPL 发展演变而来。最初的 C 语言只是为描述和实现 Unix 操作系统而提供的一种工作语言而设计的。后来，C 语言经过多次改进，先后被移植到大、中、小、微型机上，并被广泛的应用到教学、生产和科学实验等领域。现在的 C 语言是世界上应用最广泛的计算机语言之一。

以 1978 年发表的 Unix 第 7 版中的 C 编译程序为基础，Brian W.Kernighan 和 Dennis M.Ritchie 两人合著了影响深远的名著 *The C Programming Language*，这本书中介绍的 C 语言成为后来使用的 C 语言版本的基础，它被称为标准 C。1983 年，美国国家标准化协会（ANSI）制定了新的 C 语言标准，称为 ANSI C。1987 年，ANSI 公布了新的标准，称为 87 ANSI C。1990 年，国际标准化组织 ISO 接受 87 ANSI C 为 ISO C 的标准。目前流行的 C 编译系统都是以它为基础的。

1.2 C 语言的程序结构

先看两个例子，然后再来分析 C 语言的程序结构。

例 1-1 最简单的 C 程序。

实例代码如下。

```
main ()
{
    printf ("Hello Word.\n");
}
```

上面是一个简单的 C 程序，其执行结果是在屏幕上输出一行文字：



```
Hello Word.
```

其中，`main ()` 是“主函数”，每一个 C 程序都必须包含一个主程序，它是程序执行时被调用的第一个函数，是程序执行的起点。接着是“程序体”，它是两个大括号 `{}` 中间的部分。程序体是程序执行的内容，用来完成具体的工作。本例中，程序体的作用就是用来输出一行文字：“Hello Word.”。在主函数中可以用来调用其他函数：可以是系统定义的函数，也可以是用户自定义的函数。本例中用到了一个系统定义的函数：`printf ()`，其作用是将双引号内的字符串原样输出。`\n` 是换行符，其作用是在输出 Hello Word. 后换行。语句最后有一个分号，它表示本行语句结束。

下面看一个稍微复杂一点的例子。

例 1-2 在主函数中调用用户自定义函数。

实例代码如下。

```
void _printstar ()
{
    printf ("*****");
    printf ("\n");
}

void _print1 ()
{
    printf ("***** I am _print1 *****");
    printf ("\n");
}

void _print2 ()
{
    printf ("***** I am _print2 *****");
    printf ("\n");
}

void _print3 (int status)
{
    printf ("Status = %d",status);
    printf ("\n");
}

main ()
{ int status;

    printf (" Status = (1 or 2) ");
    scanf ("%d",&status);

    if (status == 1)
    {
        _print1 ();
        _print3 (status);
        _printstar ();
    }
}
```

```
    }  
    if (status == 2)  
    {  
        _print2 ();  
        _print3 (status);  
        _printstar ();  
    }  
}
```

在例 1-2 中，首先定义了 3 个自定义函数：`_printstar ()`，`_print1`，`_print2` 和 `_print3`。在定义这 3 个自定义函数时，在它们的前部都有一个关键词：`void`，它表示函数不返回任何类型的值。最后才写出了主函数 `main ()`。在程序中还用到了两个系统定义的函数：`printf ()` 和 `scanf ()`，由于 C 语言没有专门的输入输出语句，因此输入输出是由 `scanf ()` 和 `printf ()` 函数完成的。`printf ()` 函数前面已经讲过了。`scanf ()` 的作用是要求用户从键盘上输入一个值，以供程序使用。在主函数中还定义了一个整型变量：`status`，用它来接收用户的输入。

正如前面讲到的，程序从主函数 `main ()` 开始执行，它首先要求输入一个值，并提示该值可以是 1 或者 2。如果用户输入 1 并回车，系统将首先调用 `_print1 ()` 函数，在屏幕上显示出一行信息 `***** I am _print1 *****` 并换行；接着系统调用 `_print3 (status)` 函数，并把用户输入的 1 做为一个参数传递给函数 `_print3 (status)`，在屏幕上显示出一行 `Status = 1` 并换行；最后，系统调用 `_printstar ()` 函数，在屏幕上显示出一行 * 号来。其完整的显示结果如下

```
***** I am _print1 *****  
Status = 1  
*****
```

如果输入的是 2，读者可以自行分析它的输出结果。

本例涉及到的许多概念，如函数调用方式、变量的定义等，本书会在以后的章节陆续讲到。通过本例，可以概括的了解 C 程序的基本结构。

1.3 C 语言的开发环境

程序就是一组计算机能够识别并执行的指令。事实上计算机只能识别由 0 和 1 组成的二进制代码。而人们通常用于计算机程序设计的高级程序设计语言是为适应于人们的习惯而设计的，因此需要一种“编译程序”来将用高级程序设计语言编写的程序“翻译”成计算机能够识别并执行的二进制语言，这个过程称为“编译”。用高级语言编写的程序语句成为“源程序”，经过编译后的代码称为“目标程序”。将目标程序与系统的库函数和其他目标程序连接起来就形成了可执行的目标程序。程序开发环境的核心就是编译。

C 语言是优秀的计算机程序开发语言，它在 DOS 平台、Windows 平台和 Unix 开发平台下都有众多的开发环境。著名的有 Turbo C 开发环境、MS C 开发环境等。本书将以 Turbo C 为开发环境进行程序开发的讲述。

为了使用 Turbo C 开发环境进行程序开发，必须将 Turbo C 编译程序装入磁盘的某一



目录下，本书采用 Turbo C 2.0 开发环境，Turbo C 2.0 编译程序安装在 C:\TC2 目录下。要启动 Turbo C2.0 编译程序，只需在 DOS 提示符下键入 tc 命令即可。

C:\TC2\tc

Turbo C 2.0 开发环境使用了一系列下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化，大大方便了程序的开发，称为集成开发环境，如图 1-1 所示。

除 Edit 命令外，其他各项均有子菜单，只要按下 Alt 键加上某项中第一个字母（即大写字母），就可进入该项的子菜单中。

主屏幕顶上一行，显示下列内容：

File Edit Run Compile Project Options Debug Break/watch

例如要打开 File 菜单，只需按 Alt+F 键即可，如图 1-2 所示。在打开 File 菜单后可以按上下方向键 ↓ 或 ↑ 键选择子菜单。

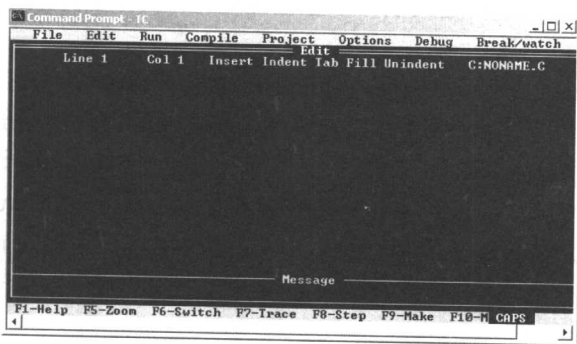


图 1-1 Turbo C 2.0 集成开发环境

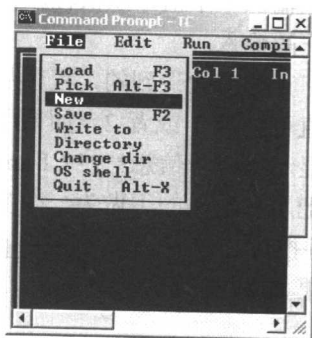


图 1-2 进入 File 菜单

Turbo C 2.0 集成开发环境中间区域为代码编辑区域，在编辑状态下可以输入和修改代码。Turbo C 2.0 集成开发环境的最下面一行是功能键提示，在这里列出了在 Turbo C 2.0 集成开发环境中最常用到的一些功能键。

在代码输入和修改完成后，根据 Turbo C 2.0 集成开发环境的功能键提示，可以按 F9 键执行程序。当按下 F9 键后，Turbo C 2.0 集成开发环境会弹出一个窗口，该窗口列出了 Turbo C 2.0 集成开发环境对当前程序的编译信息，如图 1-3 所示。此时按任意键返回到代码编辑窗口，如果有错误则光标定位于出现错误的位置。同时在屏幕的下半部分会显示错误所在的行及错误原因。根据此信息修改源代码。修改完成后再次按下 F9 键，如此反复，一直到不显示出错为止。要查看程序执行的结果，可以按 Alt+F5 键。

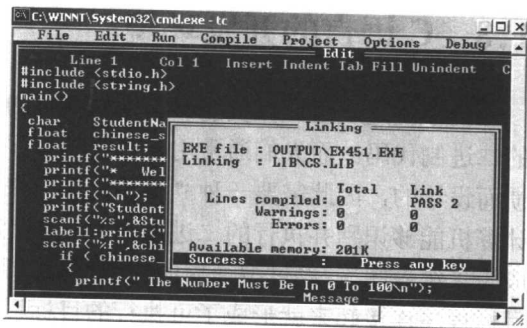


图 1-3 Turbo C 2.0 集成开发环境的编译信息

关于 Turbo C 2.0 集成开发环境的具体操作可以参阅 Turbo C 2.0 集成开发环境自带的文档 README.COM。

1.4 本章小结

本章概括的讲述了C语言的历史背景与现状、C语言的基本结构以及 Turbo C 2.0 集成开发环境的使用方法。本章中用两个实例结合 Turbo C 2.0 集成开发环境为读者讲述了C语言的编写过程,这样可以更生动、具体的让读者了解C语言的编写、调试以及编译的过程。希望读者多上机实践。

1.5 思考练习题

1. 填空题

(1) C语言是由早期的编程语言_____发展演变而来。最初的C语言只是为描述和实现_____操作系统而提供的一种工作语言而设计的。后来,C语言经过多次改进,先后被移植到_____上,并被广泛的应用到教学、生产和科学实验等领域。现在的C语言是世界上应用最广泛的_____之一。

(2) 一个C程序是从主函数_____开始执行的。

(3) 程序就是一组计算机能够识别并执行的_____。

2. 选择题

(1) C语言最初是为了描述_____而设计的。

A. Unix 系统

B. DOS 系统

C. Windows 系统

(2) 一个C程序是从_____开始执行的。

A. main 函数

B. 用户自定义函数

C. 程序文件中的第一个函数

(3) 下列哪一种开发环境不是C语言的开发环境_____。

A. MS C

B. Turbo C

C. Borland C

D. Turbo Pascal

3. 上机题

(1) 编写一个C程序,输出以下信息:

```
*****Language C*****
                Hello Word!
*****
```

(2) 将本章的例 1-1 和例 1-2 亲自上机运行。

(3) 请编写一个C程序,将输入的字符原样输出。

4. 简答题

(1) 请思考C语言出现的历史背景以及C语言的应用领域。

(2) 请写出一个C程序的基本构成。

(3) 请列举你所知道的C语言的开发环境。

第 2 章 数据类型、运算符与表达式

本章要点

- C 语言的输入输出函数。
- 常量的应用。
- 变量的种类与应用。
- C 语言中运算符与表达式的概念与应用。

本章讲述了 C 语言的一些最基本的也是重要的概念，包括数据类型、常量、变量、运算符以及表达式。这些概念会深入到 C 语言的每一个语句，是帮助构成 C 语句的核心成分。数据类型决定了数据在计算机内存中的存在形式以及存储运算形式，是程序设计的最基本的要素。运算符是具有一定数据类型的数据所要执行的运算。

2.1 输入输出函数

为了便于理解，本书先介绍几个输入输出函数，以便于读者能够更好的理解本章中的例子。

C 语言本身没有提供任何输入输出语句，它的输入输出是由输入输出函数实现的。

2.1.1 格式化输出函数 printf ()

printf () 是格式化输出函数，它以变量的类型来得到输出结果。其格式是：

```
printf (格式控制字符串, 参数 1, 参数 2, ...);
```

其中，格式控制字符串描述各参数是怎样输出的。参数 1，参数 2 等是待输出的参数，参数可以是变量，常量和表达式。下面请看例 2-1 的例子。

例 2-1 格式化输出数值 100。

实例代码如下。

```
main ()
{
    printf (" Result1 = %d\n",100) ;
    printf (" Result2 = %f\n",100.0) ;
}
```

例 2-1 的输出结果是

```
Result1 = 100
Result2 = 100.000000
```

在例 2-1 中，%d 和%f 称为格式控制字符，其中，%d 要求输出的是一个整型数值，%f 要求输出的是一个浮点型字符。对于格式控制字符%f，还可以写成以下形式：


```
printf (" Result2 = %6.2f\n",100.0) ;
```

%6.2f 代表输出的结果总共是 6 位，其中小数部分占 2 位，整数部分占 3 位，小数点占 1 位。在不指定的情况下，结果保留 2 位小数。输出结果是

```
Result2 = 100.00
```

常用的格式控制字符见表 2-1 所示。

表 2-1 常用格式控制字符

格式控制字符	d%	o%	x%	f%	c%	s%
输出	十进制整数	八进制整数	十六进制整数	浮点数	单个字符	字符串

另外，在前面的例子中使用了转义字符\n，它表示在输出完成后换行，常见的转义字符见表 2-2 所示。

表 2-2 常用转义字符

字符	\n	\f	\r	\t
作用	换行	清屏并换页	回车	Tab 符

2.1.2 格式化输入函数 scanf ()

scanf () 和 printf () 一样，也有格式控制字符串和参数列表，所不同的是，scanf () 的参数必须是指向变量的指针，用符号&来代表。关于指针的概念本书将在以后的章节讲到。Scanf () 函数的格式是：

```
scanf (格式控制字符串, 参数 1, 参数 2, ...);
```

请看例 2-2 的例子。

例 2-2 格式化输入数值。

实例代码如下。

```
main ()
{
    int a;
    float b;
    scanf ("%d,%f",&a,&b) ;
    printf ("\n") ;
    printf (" Result1 = %d, Result2 = %f\n",a,b) ;
}
```

在上面的例子中，如果键入 1, 2.2，则输出结果是

```
Result1 = 1, Result2 = 2.200000
```

再看一个小例子。