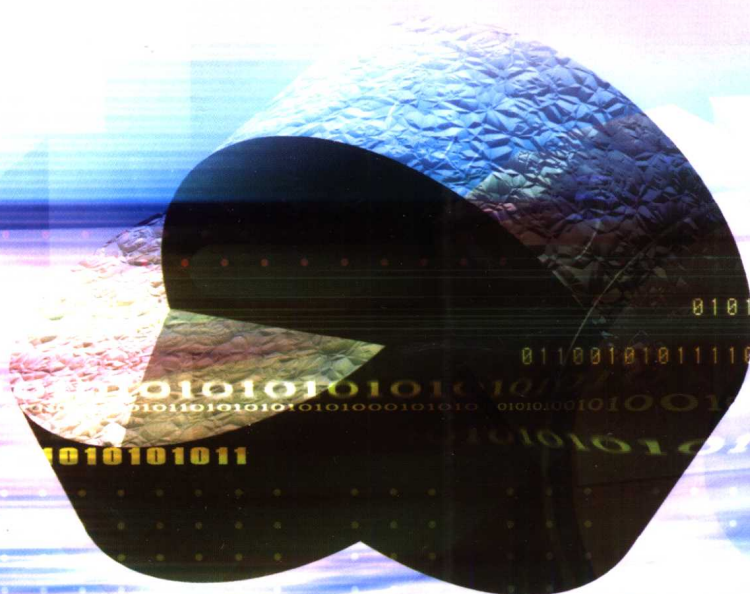


● 高等学校21世纪计算机教材

C 语言 程序设计教程

赵旻 编著



C-43

冶金工业出版社

高等学校 21 世纪计算机教材

C 语言程序设计教程

赵 旻 编著

冶金工业出版社

2003

内 容 简 介

经典的 C 语言被普遍认为是除汇编语言外程序设计语言的始祖,学习 C 语言是了解程序设计语言的必经之路。汇编语言由于其本身过于接近计算机硬件,并且较难以理解,不适合一般初学者学习。因而,学习 C 语言是学习程序设计很好的入门捷径。

本书主要介绍了 C 语言程序设计的基础知识。前 9 章介绍 C 语言最基本的语法、规则,同时逐步介绍常用的程序设计方法。随着基本知识逐渐积累,后面的章节开始出现比较复杂的程序与算法,也涉及到指针、文件操作、绘图等进阶知识。本书兼顾理论与实际,并且更强调实际操作。作者自身在学习和使用 C 语言时的一些体会与小技巧也穿插在书中,以供读者参考。

本书结构清晰、实例丰富、通俗易懂,可以作为大、中专院校计算机专业和非计算机专业的程序设计基础课程教材,也可以作为 C 语言的培训教材和自学用书。

图书在版编目(CIP)数据

C 语言程序设计教程 / 赵旻编著. —北京:冶金工业出版社, 2003.8
ISBN 7-5024-3324-4

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

中国版本图书馆 CIP 数据核字 (2003) 第 058421 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

中山市新华印刷厂有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2003 年 8 月第 1 版, 2003 年 8 月第 1 次印刷

787mm×1092mm 1/16; 15.25 印张; 368 千字; 236 页; 1-5000 册

25.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

Basic、Pascal、C 是当今比较流行的几种程序设计语言。其中，Basic 语言简单易学，但其功能不强，难以组织复杂的程序，适合刚入门的初学者使用。Pascal 语言结构严谨，较常用于描述算法，实际中也有一定影响。而 C 语言是三种语言中影响和使用最广泛的一门语言。

目前，国内的计算机资格与水平考试在程序设计方面都是以 C 语言为主，可见 C 语言的重要性。一个真正的计算机程序员应该能熟练使用 C 语言。

C 语言兼有高级语言和低级语言的特点，既可以实现算法进行数值计算，又可以直接操纵硬件，甚至可以插入汇编代码。C 语言的语法灵活、高效，使用 C 语言编写的程序通常比使用其他语言编写的程序简洁。

本书并没有介绍 C 语言的全部内容，而是重点将最基本与最常用的部分详细介绍。读者掌握了这些基础知识后，已经有能力使用 C 语言编写一般的数值计算、字符处理、文件操作的程序了。

为了兼顾理论与实践，本书中所有的示例程序都经过实际测试，确保不出现错误。示例程序的运行环境是 Turbo C 2.0，并且附上了运行结果画面。在第 1 章首先介绍了 Turbo C 2.0 的使用方法，便于读者熟悉 Turbo C 2.0。对于一些实际使用中的细节问题，结合作者自身的经验，在本书也作了说明和建议。

二、本书结构

本书章节内容的安排是由浅到深，逐步推进。

前 9 章属于 C 语言的基础，其中数据类型、输入/输出、运算符、流程控制、函数、数组是重点介绍的对象。

第 10 章与第 11 章介绍的指针与结构体是 C 语言特有的，可以说是 C 语言的精华，使用 C 语言实现较复杂的算法必须熟悉指针与结构体。

第 12 章介绍了 C 语言的位运算，要求读者具有一定的有关计算机原理方面的知识，感兴趣的读者可以继续深入研究。

第 13 章介绍了 C 语言的文件操作功能。

第 14 章介绍了用 C 语言绘图的知识。

附录 A：ASCII 码表。

附录 B：C 语言常用关键字。

附录 C：C 语言的运算符优先级别及结合方式。

附录 D：各类函数。

三、本书特点

本书结构清晰、实例丰富、通俗易懂，是作者结合自己在学习和使用 C 语言过程中的经验和技巧编写而成的。在介绍理论知识的同时，书中也给出了许多具体实施的技巧和实例，总结了容易出现的错误，并适时提出建议，读者如结合每章的练习题进行练习，就能很快掌

握 C 语言程序设计的基础知识和技巧。总体来说,本书是一本兼顾理论与实践的学习 C 语言程序设计的理想教材。

四、适用对象

本书可以作为大、中专院校计算机专业和非计算机专业的程序设计基础课程教材,也可以作为 C 语言的培训教材和自学用书。

由于时间仓促,水平有限,书中错漏之处在所难免,恳请读者批评指正。读者如果有好的意见或建议,可以发 E-mail 到 service@cnbook.net,也可以登录网站 <http://www.cnbook.net>,在该网站的论坛进行探讨。此外,本书每章练习的参考答案也可在该网站免费下载。

编 者
2003 年 6 月

目 录

第 1 章 初识 C 语言	1	3.1.3 字符型常量	13
1.1 C 语言简介	1	3.1.4 字符串常量	13
1.1.1 C 语言的主要特点	1	3.2 变量	14
1.1.2 简单的 C 语言程序	1	3.2.1 关于变量	14
1.2 关于 Turbo C 2.0	2	3.2.2 整型变量	16
1.2.1 C 语言程序的调试运行环境	2	3.2.3 浮点型变量	18
1.2.2 进入 Turbo C 2.0	2	3.2.4 字符型变量	19
1.2.3 有关汉字显示与输入的问题	4	3.2.5 字符串储存的实现	20
1.3 本书的约定	4	3.2.6 枚举型变量	21
小结	4	3.2.7 变量的存储方式	21
练习一	5	小结	27
一、选择题	5	练习三	28
二、填空题	5	一、选择题	28
三、上机题	5	二、填空题	28
第 2 章 基本结构与数据输入/输出	6	三、上机题	29
2.1 C 语言的基本结构	6	第 4 章 数据格式化输入/输出与格式转换	30
2.1.1 一个简单的框架	6	4.1 数据格式化输入输出	30
2.1.2 解析 welcome.c 程序	6	4.1.1 数据输出函数 printf()	30
2.1.3 重要的规定——语句结束符号	6	4.1.2 数据输入语句 scanf()	34
2.2 C 语言的模块化结构	7	4.1.3 字符输入函数 getchar()	35
2.2.1 模块化的实现	7	4.1.4 字符输出函数 putchar()	36
2.2.2 含有两个函数的 C 语言程序	7	4.1.5 字符串输入函数 gets()	36
2.2.3 包含多个函数的 C 语言程序	8	4.2 不同数据类型的转换	37
2.3 格式输入/输出语句	8	4.2.1 自动转换	37
2.3.1 格式输出函数 printf()	8	4.2.2 强制转换	38
2.3.2 格式输入函数 scanf()	9	小结	39
2.3.3 清除屏幕指令	10	练习四	40
小结	10	一、选择题	40
练习二	10	二、填空题	40
一、选择题	10	三、上机题	40
二、填空题	11	第 5 章 运算符与表达式	41
三、上机题	11	5.1 运算符	41
第 3 章 C 语言的数据类型	12	5.1.1 一元运算符、二元运算符与 多元运算符	42
3.1 常量	12	5.1.2 运算符的优先级	42
3.1.1 整型常量	12	5.1.3 运算符的结合方向	43
3.1.2 浮点型常量	13		

5.1.4 赋值运算符	44	小结	78
5.1.5 算术运算符	44	练习七	79
5.1.6 关系运算符	46	一、选择题	79
5.1.7 逻辑运算符	47	二、填空题	79
5.1.8 条件运算符	48	三、上机题	79
5.2 表达式	49	第 8 章 函数	80
5.2.1 算术表达式	49	8.1 函数的定义、声明与调用	80
5.2.2 逻辑表达式与条件判断	49	8.1.1 函数定义	80
小结	52	8.1.2 函数声明	81
练习五	53	8.1.3 函数调用	81
一、选择题	53	8.2 带参数的函数	82
二、填空题	54	8.2.1 形式参数与实际参数	83
三、上机题	54	8.2.2 函数的数据传递	84
第 6 章 程序的流程控制	55	8.2.3 带参数的 main 函数	89
6.1 选择型结构	55	8.3 带返回值的函数	91
6.1.1 if...else...结构	55	8.3.1 带返回值函数与不带返回值 函数的区别	91
6.1.2 switch 结构	56	8.3.2 带返回值函数的使用示例	91
6.2 循环型结构	59	8.4 递归算法	92
6.2.1 for 结构	59	8.4.1 算法思想	92
6.2.2 while 结构	61	8.4.2 汉诺塔游戏	93
6.2.3 do...while 结构	62	小结	97
6.3 其他控制语句	62	练习八	97
6.3.1 break 语句	62	一、选择题	97
6.3.2 continue 语句	63	二、填空题	98
6.3.3 goto 语句	64	三、上机题	99
6.3.4 exit 语句	65	第 9 章 数组	100
6.4 算法实现与综合运用	65	9.1 一维数组	100
6.4.1 穷举法	65	9.1.1 一维数组的声明	100
6.4.2 迭代法	67	9.1.2 一维数组的访问方式	100
小结	68	9.1.3 一维数组的应用	101
练习六	69	9.2 二维数组与多维数组	101
一、选择题	69	9.2.1 二维数组的声明	102
二、填空题	70	9.2.2 二维数组的访问方式	102
三、上机题	70	9.2.3 二维数组的应用	102
第 7 章 预编译指令	72	9.2.4 多维数组	103
7.1 宏	72	9.3 字符数组	105
7.1.1 无参数的宏	72	9.3.1 字符数组的声明	105
7.1.2 带参数的宏	74	9.3.2 字符数组与字符串操作	105
7.2 包含头文件	76	9.3.3 字符数组的应用	110
7.3 条件编译	77		

9.4 数组的储存属性	111	11.2.2 结构体变量的声明	140
小结	111	11.2.3 结构体的使用	140
练习九	112	11.3 结构体与函数	141
一、选择题	112	11.3.1 结构体作为函数的参数	142
二、填空题	112	11.3.2 结构体作为函数的返回值	144
三、上机题	113	11.4 结构体与数组	146
第 10 章 指针	114	11.4.1 结构体数组的声明	146
10.1 变量、地址与指针概述	114	11.4.2 结构体数组的初始化	146
10.2 指针变量	115	11.4.3 结构体数组的使用	147
10.2.1 指针变量的声明	115	11.5 结构体与指针	147
10.2.2 指针变量的使用	115	11.5.1 指向结构体的指针	147
10.2.3 指针运算	116	11.5.2 结构体指针的运算	148
10.2.4 二级指针与多级指针	118	11.5.3 结构体数组与指针	149
10.3 指针与函数	119	11.6 结构体与链表	149
10.3.1 函数参数的传值与传址	119	11.6.1 链表的概念	149
10.3.2 指针作为函数参数	120	11.6.2 用结构体实现链表	150
10.3.3 指针作为函数返回值	120	11.6.3 动态分配内存	152
10.3.4 指向函数的指针	123	11.6.4 链表的操作	155
10.4 指针与数组	125	11.7 共用体	165
10.4.1 指向一维数组的指针	125	11.7.1 共用体概述	165
10.4.2 指向二维数组的指针	125	11.7.2 共用体的定义	165
10.4.3 指针与字符串	127	11.7.3 共用体的声明	166
10.4.4 指针数组	129	11.7.4 共用体的使用	166
10.5 使用指针的注意事项	131	11.8 枚举类型数据	166
10.5.1 指针与所指数据的类型		11.8.1 枚举类型的定义	167
不一致	131	11.8.2 枚举类型变量的声明	167
10.5.2 访问未知的地址	132	11.8.3 枚举类型变量的使用	167
10.5.3 指针级别混淆	133	小结	168
10.5.4 被调用函数擅自修改		练习十一	169
主函数变量的值	133	一、选择题	169
10.5.5 函数返回一个无效的指针	134	二、填空题	170
小结	135	三、上机题	171
练习十	136	第 12 章 位运算	172
一、选择题	136	12.1 位运算概述	172
二、填空题	137	12.2 位运算符	172
三、上机题	138	12.2.1 取反运算	172
第 11 章 结构体、共用体与枚举型数据	139	12.2.2 与运算	173
11.1 结构体概述	139	12.2.3 或运算	174
11.2 结构体变量	139	12.2.4 异或运算	175
11.2.1 结构体的定义	139	12.2.5 左移运算	175
		12.2.6 右移运算	177

12.2.7 位运算符的优先级	179	14.2 绘图	211
12.3 位运算应用实例	181	14.2.1 基本绘图函数	211
小结	185	14.2.2 绘制常见图形	214
练习十二	185	14.2.3 填充图形	218
一、选择题	185	14.3 输出文字	220
二、填空题	185	14.3.1 图形方式下输出文字	220
三、上机题	186	14.3.2 设定文字输出样式	221
第 13 章 文件	187	14.3.3 设定文字输出定位方式	221
13.1 文件概述	187	14.4 其他相关函数	221
13.1.1 文件类型	187	14.4.1 getmaxx() 与 getmaxy() 函数	222
13.1.2 文件指针	187	14.4.2 getmaxcolor() 函数	222
13.2 打开、关闭文件	187	14.4.3 getpixel() 函数	222
13.2.1 打开文件函数 fopen()	188	小结	222
13.2.2 关闭文件函数 fclose()	189	练习十四	223
13.3 顺序读写文件	190	一、选择题	223
13.3.1 输入、输出字符	190	二、填空题	223
13.3.2 输入、输出字符串	192	三、上机题	223
13.3.3 格式化输入、输出	195	附录 A ASCII 码表	225
13.3.4 以二进制方式读写文件	198	A.1 控制字符	225
13.4 定位读写文件	202	A.2 键盘字符	225
13.4.1 fseek() 函数	202	A.3 特殊字符	226
13.4.2 rewind() 函数	203	附录 B C 语言常用关键字	227
13.4.3 ftell() 函数	203	B.1 数据声明关键字	227
13.5 目录操作	204	B.2 数据储存类别关键字	227
13.5.1 改变当前盘	204	B.3 命令语句	227
13.5.2 改变当前目录	205	B.4 内部函数	228
13.5.3 创建目录	206	附录 C C 语言的运算符优先级及	
13.5.4 删除目录	206	结合方式	229
小结	207	附录 D 各类函数	230
练习十三	207	D.1 输入/输出函数	230
一、选择题	207	D.2 数学函数	232
二、填空题	208	D.3 字符函数	233
三、上机题	208	D.4 字符串函数	234
第 14 章 C 语言绘图	209	D.5 内存分配函数	234
14.1 图形模式	209	D.6 其他函数	234
14.1.1 打开、关闭图形模式	209	D.7 图形函数	235
14.1.2 图形参数设定	210		

第 1 章 初识 C 语言

本章作为全书的第 1 章，主要目的是使读者对 C 语言有个整体的认识。本章同时也介绍了 C 语言的编译环境 Turbo C 2.0 的基本使用方法（本书中所有示例程序都将在 TurboC 2.0 环境中运行），最后是有关在程序代码中插入中文的问题以及本书示例代码的约定，这些细节问题在后面章节中将不再赘述。

1.1 C 语言简介

下面首先介绍了 C 语言的主要特点，然后给出了一个结构简单的 C 语言程序。

1.1.1 C 语言的主要特点

C 语言是世界上最流行的几种程序设计语言之一。与其他语言相比，C 语言具有以下主要特点：

- （1）语句简洁、紧凑，使用灵活、方便。
- （2）生成代码的运行效率高。
- （3）允许直接对内存物理地址的数据操作，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。

由于 C 语言的上述特点，使其适合编写各种系统软件和应用软件，深受广大程序员的欢迎。C 语言也是一种经典的结构化程序设计语言，是编程学习者不可不学的内容。

1.1.2 简单的 C 语言程序

C 语言源程序的文件后缀名为.c，经过编译后可以生成.exe 的可执行文件，脱离 C 语言编译器运行。这里先给出一个简单的 C 语言程序的源代码，以使读者对 C 语言有一个初步的认识。

【例 1-1】

```
welcome.c
main( )
{
    printf("Welcome to C language world!\n");
}
```

以上程序在 Turbo C 2.0 上运行的结果，如图 1-1 所示。

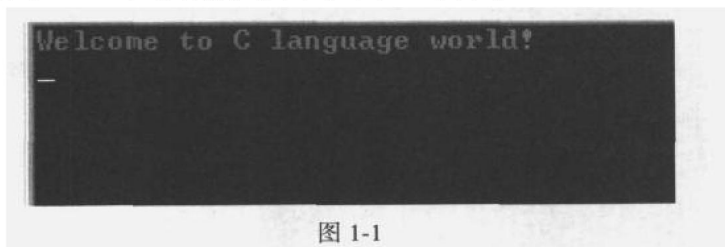


图 1-1

它的功能是在屏幕上打印 Welcome to C language world 及一个回车符。

读者在这里无需知道全部代码指令的细节，只要初步了解 C 语言程序的基本结构就可以

了。其中第一行 `main()` 定义了一个主函数，而第二行和第四行的 `{}` 花括弧对之间的内容就是主函数的函数体。上面的函数体只有一行，就是调用一个屏幕输出语句 `printf()` 打印出“欢迎来到 C 语言世界”的英文加一个回车符。

C 语言程序基本结构将在下一章介绍。

1.2 关于 Turbo C 2.0

Turbo C 2.0 是一个 C 语言的编译器。写好的 C 语言程序代码需要编译后才能运行。

将由字符组成的源程序代码转化为计算机可以直接执行的二进制代码的过程就是编译。编译原理对于初学者来说过于复杂，甚至许多程序员也未必能够很好理解。读者目前只需要清楚编译的概念和作用，最重要是学好 C 语言本身。

编译器的工作是：对代码进行语法检查，如果有错误则给出错误信息；然后进行文件链接（引入需要的文件）；最后运行程序并输出结果。Turbo C 2.0 还可以生成 .exe 可执行文件，使程序脱离 Turbo C 2.0 环境运行。Turbo C 2.0 同时也是 C 语言源程序的编辑、调试环境，它不仅包含了文本编辑的必要功能，也提供了许多辅助程序设计者进行代码查错、监控程序运行的工具。熟悉使用 Turbo C 2.0 将使编程效率大大提高。

1.2.1 C 语言程序的调试运行环境

在上节中的 C 语言程序是在 Turbo C 2.0 环境下调试运行的。本书所有程序均使用 Turbo C 2.0 作为调试运行环境。它遵循 ANSI C 的标准，在 Turbo C 2.0 里编写的 C 语言代码也能在其他遵循 ANSI C 标准的 C 语言编译器下运行。

1.2.2 进入 Turbo C 2.0

1. 启动方式

在 Turbo C 2.0 的安装目录下，运行 `tc` 进入 Turbo C 2.0 的主界面。

2. 运行环境

Turbo C 2.0 可以在纯 DOS 方式下，或在 Windows 的 DOS 方式下运行。

3. 主界面

Turbo C 2.0 的主界面如图 1-2 所示。



图 1-2

4. 编辑调试程序与常用操作

进入主界面后, 当前状态是文本编辑状态, 可以立刻输入或编辑 C 语言程序代码。屏幕上部的是菜单条, 菜单条上有多个下拉菜单, 通过按 Alt+子菜单标题首字母 (红色) 打开菜单。如要打开 File 菜单, 可以按 Alt+F 组合键。打开一个菜单后, 可以使用方向键选择打开其他的子菜单或功能, 按 ESC 键关闭菜单并回到文本编辑状态。

菜单中提供的常用操作如下:

1) File 菜单

Load: 打开.c 文件。

New: 建一个新文件。

Save: 保存修改结果。

Write To: 另存为一个新文件。

OS Shell: 切换到 MS-DOS 状态 (用 Exit 命令返回 Turbo C 2.0)。

Quit: 退出 Turbo C 2.0。

2) Run 菜单

Run: 运行程序。

Program Reset: 全部终止, 程序重新开始运行, 快捷键为 Ctrl+F9。

Step Over: 逐行运行程序代码, 快捷键为 F8, 按 F8 键执行下一行。

User Screen: 查看运行输出结果。

3) Compile 菜单

Compile To Object File: 链接成目标文件。

Make EXE File: 编译成可执行程序 (无需在 Turbo C 2.0 下就可以直接运行, 如图 1-3 所示)。生成的 exe 文件与 C 语言程序文件同名, 存放于 Turbo C 2.0 所在目录下。

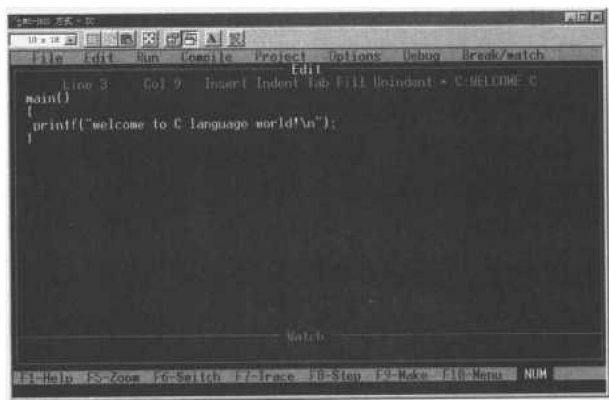


图 1-3

将例 1 的 welcome.c 输入编辑区后如图 1-3 所示。要运行程序按 ALT+R 呼出运行菜单选 Run 或者按组合键 Ctrl+F9, 将会弹出一个正在编译窗口。如果程序没有错误, 过一会窗口消失并返回编辑状态。如果程序有错误则在弹出窗口中列出错误数目, 并在 Message 区域显示出错信息。大多数错误信息会标明错误所在的行数以及错误类型, 可以很方便地根据错误信息修改代码。

查看程序运行结果按 Alt+F5。使用 F6 键切换窗口。

编辑窗口左上方显示了当前光标所在行、列位置以及插入状态,右上方显示了 C 语言程序的文件名。

屏幕下方是常用快捷键说明以及小数字键盘开关状态 NUM。

若新建一个 C 语言文件,默认的名称为 noname.c,首次保存时系统会提示更改文件名。每次不带参数启动时 Turbo C 2.0 时,系统自动打开 noname.c 文件。若带参数启动 Turbo C 2.0,则打开参数所指的 C 语言程序。如:

```
C:\>tc welcome.c
```

也可以省略文件扩展名,如:

```
C:\>tc welcome
```

1.2.3 有关汉字显示与输入的问题

下面介绍一下在 Turbo C 2.0 中显示和输入汉字的问题。这不属于学习 C 语言的内容,仅仅为读者提供一些方便。Turbo C 2.0 本身不支持中文汉字,因而在 Turbo C 2.0 中使用汉字可能会带来一些无法预料的错误。为了避免不必要的麻烦,在用英文可以表达的情况下,不建议在 Turbo C 2.0 中使用中文。尽管如此,还是可以通过一些方法在 Turbo C 2.0 中实现中文汉字的显示和输入。

1. 纯 DOS 方式下

在纯 DOS 方式下可以先运行中文环境(如 UC DOS 7.0),然后进入 Turbo C 2.0,就可以显示中文。在需要打汉字时用热键调出输入法输入中文即可。

2. Windows 界面下

Windows 的 DOS 窗口本身就可以显示中文,只需按上方的“汉”按钮即可。要输入中文,在 Windows 的 MS-DOS 方式下先运行 PDOS95 命令,然后再进入 Turbo C 2.0,用 Ctrl + 空格键调出输入法输入中文。

注意:在程序中的中文只可以作为提示信息,决不能作为变量名和函数名使用!

1.3 本书的约定

本书中的程序段中在/*和*/之间的中文是为向读者说明某些代码的作用,在程序中没有任何实际作用。/*与*/为注释标识,在/*与*/之间的所有内容编译器将不予理会。例:

```
printf("Welcome to C language world!\n"); /*这里是说明*/
```

```
printf("Welcome to C language world!\n");
```

上面两行代码的作用完全一样。读者在实践编程时不必将注释连同程序代码一起输入。

小结

(1) C 语言是一门经典的程序设计语言,有灵活、高效、功能强大等特点,深受程序设计者欢迎。

(2) 由短短几行 C 语言代码就可以构成一个简单的 C 语言程序。

(3) 本书中使用 Turbo C 2.0 运行 C 语言程序。Turbo C 2.0 是基于 DOS 的,也能够在 Windows 环境下运行。

(4) Turbo C 2.0 本身不支持中文输入输出,要在 Turbo C 2.0 中使用中文,需要先加载

中文环境。因为很可能引起错误，所以一般情况下不提倡在 C 语言程序代码中出现中文。

(5) 本书中的示例程序中用/*和*/括起的内容是程序代码的注释,没有任何实际作用,不必与代码一同输入。

练习一

一、选择题

1. 下列各种程序设计语言中，使用最广泛的一种是()。
- A. Basic B. Pascal
- C. FORTRAN D. C
2. 以下()编译器可以用于编译C语言程序。
- A. Turbo Pascal B. MASM
- C. Turbo C 2.0 D. QBasic

二、填空题

1. C 语言源程序的后缀名是_____，最终生成的可执行文件的后缀名是_____。
2. 在 Turbo C 2.0 中，运行当前程序的快捷键是_____；查看输出结果的快捷键是_____。
3. C 语言程序中注释符号是_____。
4. 下面程序的输出结果是_____。

```
main ( )
{
    printf("Hello!");
    printf("Mr. Brown!");
}
```

三、上机题

1. 在 TurboC 2.0 中将 welcome.c 的代码输入, 验证运行结果。
2. 将 welcome.c 程序编译成 welcome.exe 文件, 在 DOS 或 Windows 环境下直接运行, 验证运行结果。

第2章 基本结构与数据输入/输出

本章的目的是让读者能看懂简单的 C 语言程序，这样在学习以后章节时看示例程序不至于困惑。

2.1 C 语言的基本结构

编写 C 语言程序，无论程序长短，都要遵循最基本 C 语言程序规则。

2.1.1 一个简单的框架

一个最简单的 C 语言程序结构框架如下：

```
main()  
{  
... /*函数体*/  
}
```

第一行的 main 是主函数名。所有的 C 语言程序的主函数名都是 main。组成 C 语言的基本元素是函数。每个完整的 C 语言程序都必须有一个主函数，程序的执行先以主函数开始，再由它调用其他的函数。

一般情况下 main 函数没有参数，但小括号 () 必须写出。对于其他函数也一样，即使没有参数也要写小括号。

函数体以左花括弧 { 开始，以右花括弧 } 结束，函数体里面可以执行指令和调用函数，也可以什么也没有，此时整个程序将不做任何事情，但它是一个合法的 C 语言程序。

2.1.2 解析 welcome.c 程序

仍然以本书开始的 welcome.c 程序为例：

【例 2-1】

```
main() /*主函数声明*/  
{ /*函数开始*/  
printf("Welcome to C language world!\n"); /*函数体*/  
} /*函数结束*/
```

main 函数体内有一条 printf("Welcome to C language world!\n"); 语句，于是程序的运行结果就是运行这条语句所得的结果。

2.1.3 重要的规定——语句结束符号

在 C 语言中，函数体内的全部指令都要在末尾加上一个分号表示命令完结。这是 C 语言的一个重要规定，必须无条件遵循。即使函数体内只有一条指令，也必须加上分号结尾。

值得注意的是，初学者在编写程序时，大多数的错误都是由于漏了指令结尾的分号而造成的。

消掉 welcome.c 中第 3 行末尾的；再运行则产生下面的出错信息：

```
Error C:\TC\WELCOME.C 3: Statement missing ; in function main  
表示在 welcome.c 的程序第 3 行的指令缺少；结尾。
```

2.2 C 语言的模块化结构

程序设计的模块化是现今的发展趋势。其主要思想是把一个大型的程序分成多个功能模块，各自实现一定的功能。一个模块相对其他模块是独立的，互不干扰。模块和模块之间可以互相调用，并通过参数传递数值。每个模块都可以共享公共数据，但不能操作其他模块的私有数据。

2.2.1 模块化的实现

C 语言通过各个函数实现模块化的程序设计。一个函数相当于一个模块。每个函数完成一定功能，各个函数互不干扰。

一个函数可以调用其他一个或多个函数，甚至可以调用自己本身。函数间通过参数和变量传递数据，函数也可以返回数值。

2.2.2 含有两个函数的 C 语言程序

下面看看有两个函数的 C 语言程序：

【例 2-2】

```
welcome2.c
main( )
{
    void welcome( );           /*第3行  声明 welcome 函数*/
    welcome( );                /*第4行  调用 welcome 函数*/
}

void welcome( )                /*第6行  定义 welcome 函数*/
{                               /*welcome 函数开始*/
    printf("I am in fuction welcome.\n"); /*屏幕打印语句*/
    printf("Welcome to C language world!\n"); /*屏幕打印语句*/
}                               /*welcome 函数结束*/
```

这个程序除主函数 main 外，还有一个 welcome 函数。主函数 main 中调用了 welcome 函数，运行结果如图 2-1 所示。

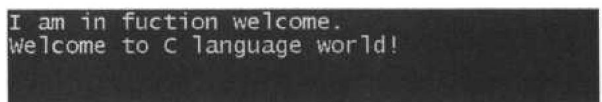


图 2-1

程序的第 3 行是 welcome 函数声明，告诉 main 函数下面可以调用 welcome 函数。如果被调用函数在调用函数之前已经定义，则可以不声明而直接调用。但最好把所有被调用函数都声明一次，便于修改和维护程序。

所有声明语句，包括函数声明和下面将学到的变量声明在函数体中的位置必须在执行语句之前。否则声明语句无效，当要调用函数或变量时将产生错误。执行语句即产生动作的语句，如数据输入输出，变量赋值等。

第 4 行调用了 welcome 函数，注意尽管 welcome 函数没有参数，但是还要带上 () 结尾。

第 6 行的作用是定义一个函数，名称为 welcome，后面 {} 内是 welcome 的函数体。

注意：第 6 行定义函数的末尾没有分号，因为它不是函数体内的语句，而第 8、9 行语

句末尾则需要分号结尾。

函数定义和声明中出现的 void 类型说明符, 它表示 welcome 函数没有返回值。与 Pascal 语言相比, C 语言中没有子过程 (或称子程序), 没有返回值的函数其作用就相当于子过程。

2.2.3 包含多个函数的 C 语言程序

多个函数的结构与 welcome2.c 程序的结构类似。

应该尽量把功能相似的函数放在一起, 以便于集中管理, 提高编程效率。

有关函数的更多知识将在第 8 章详细介绍。

2.3 格式输入/输出语句

输入输出是程序与用户交流的方式, 因而学习好输入输出语句意义重大。本小节只是简单介绍输入输出语句, 以便读者看懂示例程序。有关数据输入输出更深入的知识, 将在第 4 章详细介绍。

2.3.1 格式输出函数 printf() ·

再来看看熟悉的 welcome.c 程序:

```
main()  
{  
    printf("Welcome to C language world!\n");    /*屏幕打印语句 printf( )*/  
}
```

对于这个程序的结构相信读者已经清晰, 下面来分析一下使用过多次的屏幕打印语句 printf ()。

(1) 首先明确 printf () 也是一个函数, 但这个函数无需由用户定义, 也无需声明, 它是 C 语言的内部函数, 可以随时调用。

(2) 在 () 内是 printf 这个函数的参数表, 在上面的程序中有一个参数, 就是 "Welcome to C language world!\n"。其中 Welcome to C language world! 是一个字符串, 就是要打印到屏幕上的内容, 它作为 printf () 函数的参数时要用双引号括起来, 说明它是一整串字符。

(3) 在语句中出现的 \n 是一个转义字符, 即代表另外一种操作, 而不是要输出代表字符串 "\n"。在 C 语言的格式输出函数 printf () 函数中, 规定以 % 开头的和 \ 开头的若干个字符分别为格式字符和转义字符。格式字符用于规定变量的输出格式, 转义字符用于完成诸如输出换行或跳到下一个输出区域等特殊操作。

由于到目前为止还没有接触过 C 语言变量的数据类型, 所以格式字符的含义将在学习了 C 语言的数据类型之后详细介绍, 下面列出常用的转义字符及其含义:

\n: 输出换行符。后面的内容将输出在下一行。

\t: 跳到下一个输出区域。DOS 状态下一行可以输出 80 个字符, 把一行分为 10 个输出区域, 每个区域占 8 个字符。如果在当前输出区域超过 7 个字符, 那么遇到 \t 则将跳到下一个输出区域, 因为 \t 至少要跳开一个字符位置的空格。如果已经到第 10 个输出区域则跳到下一行第 1 个区域。

\b: 退格符。遇到这个转移符, 光标将向左移动一格。若移动后光标所在位置已经有内容, 则 \b 后面的输出将覆盖原有的内容。若是输出到打印机, 利用它可以实现两个字符打印