

从零开始

—— C 语言程序设计基础培训教程

老虎工作室 孙海侠 赖积滨 姜继红 编著



人民邮电出版社

内 容 提 要

本书详细介绍了 C 语言的数据类型、变量、各种基本语句和函数的功能、数组、指针、结构体、位运算、文件操作以及程序设计的方法，是学习 C 语言的入门教程。

本书通俗易懂，充分考虑了初学者的实际需要，真正“从零开始”，可以使对 C 语言一点都不懂的读者，通过学习本书而掌握 C 语言的应用方法和技巧。本书每一章的最后都有相应的练习题，可帮助读者检验学习效果。

为了方便读者学习，书中用到的程序代码已经上载到老虎工作室网站 (<http://www.laohu.net>)，读者可以根据需要下载使用。

本书既可以作为各类专业技术人员学习 C 语言程序设计的培训教材，也可作为相关专业大、中专及本科生的教学参考书。

从零开始——C 语言程序设计基础培训教程

◆ 编 著 老虎工作室 孙海侠 赖积滨 姜继红
责任编辑 李永涛
执行编辑 徐宝姝

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692

北京汉魂图文设计有限公司制作
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092 1/16
印张：20
字数：482 千字 2004 年 6 月第 1 版
印数：1— 000 册 2004 年 6 月北京第 1 次印刷

ISBN 7-115-12341-1/TP · 4019

定价： .00 元

本书如有印装质量问题，请与本社联系 电话：(010) 67129223

第 1 章 C 语言概述	1
1.1 C 语言的诞生与发展	1
1.2 C 语言的特点	2
1.3 C 程序的基本结构	3
1.4 建立与运行 C 程序的过程	8
1.5 用 Turbo C 运行 C 程序的上机步骤	9
1.6 小结	13
1.7 习题	14
第 2 章 数据类型	15
2.1 常量	15
2.1.1 字面常量	15
2.1.2 符号常量	18
2.2 变量	19
2.2.1 变量的定义	19
2.2.2 变量的初始化	24
2.3 小结	24
2.4 习题	24
第 3 章 运算符、表达式和语句	27
3.1 运算符和表达式	27
3.1.1 赋值运算符和赋值表达式	28
3.1.2 算术运算符和算术表达式	29
3.1.3 逗号运算符和逗号表达式	34
3.1.4 求字节数运算符和负值运算符	35
3.2 语句	36
3.2.1 空白对语句的影响	36
3.2.2 表达式语句	36
3.2.3 复合语句	37
3.3 小结	38

3.4 习题	38
第 4 章 数据的输入与输出	41
4.1 格式输出函数 printf()和格式输入函数 scanf()	41
4.1.1 printf()函数.....	41
4.1.2 scanf()函数.....	48
4.2 字符输出函数 putchar()和字符输入函数 getchar()	52
4.2.1 putchar()函数.....	52
4.2.2 getchar()函数.....	53
4.3 小结	54
4.4 习题	54
第 5 章 顺序结构程序设计	57
5.1 程序设计	57
5.1.1 算法.....	58
5.1.2 结构化程序设计.....	60
5.2 类型转换	63
5.2.1 自动类型转换.....	63
5.2.2 强制类型转换.....	66
5.3 顺序结构程序举例	68
5.4 小结	70
5.5 习题	71
第 6 章 选择程序结构设计	73
6.1 关系运算符和关系表达式	73
6.2 逻辑运算符和逻辑表达式	75
6.3 if 语句.....	77
6.3.1 if 语句的 3 种形式	77
6.3.2 选择语句的嵌套.....	82
6.3.3 条件运算符和条件表达式.....	85
6.4 switch 语句.....	87
6.5 程序举例	91
6.6 小结	94
6.7 习题	94
第 7 章 循环结构程序设计	97
7.1 while 语句.....	97

7.2 for 语句	100
7.3 do-while 语句	104
7.4 循环的嵌套	106
7.5 循环的辅助语句	110
7.6 goto 语句	114
7.7 两种常用的算法	116
7.8 程序举例	118
7.9 小结	120
7.10 习题	121
第 8 章 数组	123
8.1 数组的定义和引用	123
8.1.1 数组的定义	123
8.1.2 数组的引用	124
8.2 数组元素在内存中的存储方法	126
8.3 数组的初始化	127
8.3.1 一维数组的初始化	127
8.3.2 二维数组的初始化	128
8.4 数组应用举例	129
8.5 字符数组	135
8.5.1 字符数组的定义和初始化	135
8.5.2 字符数组的引用	136
8.5.3 字符串	137
8.5.4 字符数组的输入和输出	138
8.5.5 字符串处理函数	140
8.5.6 应用举例	142
8.6 小结	144
8.7 习题	144
第 9 章 指针	147
9.1 地址和指针	147
9.2 指针的运算	148
9.2.1 指针的定义	148
9.2.2 使指针指向对象的方法	149
9.2.3 取地址运算符和访问地址运算符	150
9.2.4 指针的引用	151
9.2.5 指针的运算	152
9.3 指针和数组	154

9.3.1 指针与一维数组	154
9.3.2 指针与二维数组	156
9.4 指针和字符串	160
9.5 指针数组和二级指针	164
9.5.1 指针数组	164
9.5.2 二级指针	166
9.6 小结	168
9.7 习题	168
第 10 章 函数	171
10.1 概述	171
10.2 函数的定义	172
10.3 函数的调用	173
10.3.1 函数调用的定义和声明	173
10.3.2 函数的返回	177
10.4 变量作用域	179
10.4.1 局部变量	179
10.4.2 全局变量	180
10.5 变量的存储类型	182
10.6 函数间数据传递方式	190
10.6.1 函数间的参数传递	190
10.6.2 函数返回值	195
10.6.3 全局变量	197
10.7 函数的嵌套调用	198
10.8 函数的递归调用	200
10.9 函数指针和函数指针数组	204
10.9.1 函数指针	204
10.9.2 函数指针数组	208
10.10 main()函数的参数和返回值	209
10.11 内部函数和外部函数	211
10.12 小结	212
10.13 习题	212
第 11 章 预处理	215
11.1 宏定义	215
11.1.1 不带参数的宏定义	215
11.1.2 带参数的宏定义	218
11.2 文件包含	220

11.3 条件编译	222
11.4 小结	226
11.5 习题	227
第 12 章 结构体与共用体	229
12.1 结构体类型的声明	229
12.2 结构体类型的变量	230
12.2.1 结构体类型变量的定义	230
12.2.2 结构体类型变量的初始化	233
12.2.3 结构体类型变量的引用	234
12.3 结构体数组	235
12.4 指向结构体变量和结构体数组的指针	239
12.5 结构体与函数	242
12.5.1 结构体在函数间的传递	242
12.5.2 结构体类型函数或结构指针型函数	246
12.6 链表	248
12.6.1 链表的基本知识	248
12.6.2 动态存储分配函数	249
12.6.3 链表的操作	250
12.7 共用体	257
12.7.1 共用体类型的声明	257
12.7.2 共用体类型的变量	258
12.7.3 程序举例	260
12.8 枚举类型	262
12.9 用 typedef 定义类型	265
12.10 小结	267
12.11 习题	268
第 13 章 位运算	271
13.1 二进制数、位和字节	271
13.2 位运算符和位运算	272
13.3 程序举例	276
13.4 位段	277
13.5 小结	279
13.6 习题	279

第 14 章 文件	281
14.1 文件概述	281
14.2 文件类型指针	282
14.3 文件的打开与关闭	283
14.3.1 文件的打开	283
14.3.2 文件的关闭	285
14.4 文件读写函数	285
14.4.1 字符读写函数	286
14.4.2 字符串读写函数	288
14.4.3 格式化读写函数	290
14.4.4 数据块读写函数	290
14.5 文件检测函数	292
14.6 文件的定位	294
14.7 文件管理函数	297
14.8 小结	298
14.9 习题	298
附录 常用资料	301
F.1 ASCII 码	301
F.2 C 运算的优先级和结合性	302
F.3 常用的 C 语言库函数	303

第1章 C 语言概述

C 语言是国际上广泛流行的计算机高级语言，既可以用来编写系统程序，也可以用来编写不同领域的应用程序。现在，C 语言编译系统几乎可以在所有的操作系统，以及各种类型的计算机上运行。从个人计算机到大型计算机，有相当大比例的软件是用 C 语言编写的。C 语言突出的优点备受计算机爱好者及广大学生的青睐。

1.1 C 语言的诞生与发展

在 C 语言诞生以前，系统软件主要是用汇编语言编写的，包括 UNIX 操作系统在内。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差，但一般的高级语言又难以实现对计算机硬件的直接操作，而这正是汇编语言的优势。人们盼望有一种兼有汇编语言和高级语言特性的新语言，于是，C 语言就在这种情况下应运而生了。

C 语言的祖先是 ALGOL 60 (ALGOrithmic Language 60)。ALGOL 60 是 1960 年由国际计算机委员会设计的一种面向过程的高级语言。它是一种结构化语言，用它编写的程序具有可读性和可移植性好的特点。但是，它不能直接对硬件进行操作，不宜用来编写系统程序。

1963 年，英国的剑桥大学和伦敦大学首先将 ALGOL 60 发展成 CPL (Combined Programming Language) 语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些，但规模比较大，难以实现。

1967 年，英国剑桥大学的 Martin Richards 将 CPL 改为 BCPL (Basic Combined Programming Language) 语言。BCPL 语言比 CPL 语言大为简化。

1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，又做了进一步简化，设计出了很简单且很接近硬件的 B 语言，并用 B 语言写了第一个 UNIX 操作系统，1971 年在 PDP-7 小型机上实现。但 B 语言过于简单，功能有限。

1972 年，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言既保持了 BCPL 和 B 语言的精练和接近于硬件的特点，也克服了它们过于简单、数据无类型等缺点。1973 年，Ken Thompson 和 D.M.Ritchie 再次合作将 1969 年他们用汇编语言编写的 UNIX 操作系统改用 C 语言编写，即 UNIX 第 5 版。其中 C 语言代码占 90% 以上，只保留了少量汇编语言代码。这样使 UNIX 操作系统向其他类型的机器上移植变得相当简单。到了 20 世纪 70 年代中期，UNIX 操作系统和 C 语言作为软件设计师的良好工具传遍了贝尔实验室，接着也传遍了几乎所有的美国大学校园，UNIX 和 C 开始风靡世界。

UNIX 操作系统的广泛使用，促进了 C 语言的迅速发展及普及，C 语言的发展与普及也促进了 UNIX 操作系统的推广。1978 年，出现了独立于 UNIX 和 PDP 计算机的 C 语言，从而，C 语言被迅速移植到大、中、小型机及微型机上。同年，以 UNIX 第 7 版中的 C 编译程序为基



础, Brian W.Kernighan 和 Dennis M.Ritchie (合称 K&R) 合著了影响深远的《The C Programming Language》一书。这本书介绍的 C 语言成为后来广泛使用的 C 语言版本的基础, 它被称为标准 C。随着微型计算机的日益普及, 出现了许多 C 语言版本。由于没有统一的标准, 使得这些 C 语言之间出现了一些不一致的地方。从 1983 年开始, 为了改变这种情况, 美国国家标准研究所 (ANSI) 根据 C 语言问世以来各种版本对 C 语言的发展和扩充, 经过长达 5 年的努力, 制定了 ANSI C 标准。1988 年, K&R 修改了他们的经典著作《The C Programming Language》, 按 ANSI C 标准重新编写该书。现在一般称 ANSI C 为新标准, 称 K&R C 为旧标准。本书以 ANSI C 新标准来介绍。

C++语言是 C 语言发展的新阶段, 是应用广泛的面向对象的程序设计语言。

目前, 在微机上广泛使用的 C 语言编译系统有 Microsoft C、Turbo C、Borland C 等。虽然它们的基本部分都是相同的, 但还是有一些差异, 所以请读者注意自己所使用的 C 编译系统的特点和规定 (参阅相应的手册)。Turbo C 2.0 完全支持新标准, 并提供集成开发操作环境。本书选定的上机环境是 Turbo C 2.0。

1.2 C 语言的特点

C 语言发展如此迅速, 而且成为最受欢迎的语言之一, 主要因为它具有强大的功能。许多著名的系统软件, 如 dBASE III PLUS、dBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序, 就更能显示 C 语言的优势了, 如 PC-DOS3.0、WordStar 等就是用这种方法编写的。归纳起来 C 语言具有以下特点。

(1) C 语言同时具有汇编语言和高级语言的优势

C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作, 而这 3 者是计算机最基本的工作单元。

(2) 语言简洁、使用灵活, 便于学习和使用

C 语言在表示方法上力求明了易懂、简单易行。例如, C 语言的书写格式没有限制, C 语言提供了预处理功能以及大量而有效的函数库等。

(3) 运算符类型丰富

C 语言的运算符包含的范围很广泛, 从而使 C 语言的运算类型极其丰富, 表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(4) 数据类型丰富, 并有多种数据结构

C 语言具有各种各样的数据类型, C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等, 能用来实现各种复杂数据结构的运算, 尤其引入了指针概念, 可使程序效率更高。

(5) 同汇编语言相比, 用 C 语言写的程序可移植性好

C 语言的一个突出的优点就是适合于在多种操作系统以及各种类型的计算机上运行, 因此移植起来很方便。

(6) 结构化程序设计语言

结构化程序设计语言的显著特点是代码和数据分隔化, 即程序的各个部分除了必要的信息交流外彼此独立。这种结构化设计的方式可使程序层次清晰, 便于使用、维护以及调



试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多种循环语句、条件语句来控制程序的流向，从而使程序完全结构化。

(7) 生成的目标代码质量高，程序执行效率高

高级语言能否用来描述系统软件，特别是操作系统、编译程序等，除要求语言表达能力强之外，很重要的一个因素是语言生成的目标代码质量如何。许多实验表明，针对同一个问题，用 C 语言编写的程序，其目标代码的效率仅比汇编语言低 10%~20%。由于用高级语言描述问题比汇编语言编程迅速，工程量小，可读性和可移植性好，易于调试和修改，所以 C 语言成为描述系统软件和应用软件比较理想的工具。

(8) 程序设计较自由，语法限制不严格

对数组下标越界不做检查，对变量的类型使用比较灵活，C 语言允许程序编写者有较大的自由度，因此放宽了语法检查。

C 语言具有上述诸多优点，因而具有旺盛的生命力，但也有一些缺点和不足。如运算符的优先级太多，不便于记忆，类型转换比较随便，检测手段太弱等，这就要求程序员对程序设计更熟练、查错能力更强。但是，瑕不掩玉，多种程序设计的应用已经证明 C 语言是一种极为有效且表达能力很强的语言。

1.3 C 程序的基本结构

用 C 语言编写的程序，称为 C 语言源程序，简称为 C 程序。C 程序一般由一个或多个函数组成，这些函数可放在同一个文件中，也可分散放在几个文件中。

一、一个简单的例子

先看一个简单的例子，认识一下 C 程序的构成。

【例1-1】 一个计算梯形面积的 C 程序。

```
/*求梯形面积*/
#include <stdio.h>           /*文件包含*/
main()
{
    int  up,bot,high;        /*定义了上底、下底和高为整型变量*/
    float area;              /*定义了面积为实型变量*/
    up=3;                    /*上底等于 3*/
    bot=5;                   /*下底等于 5*/
    high=4;                  /*高等于 4*/
    area=(up+bot)*high/2.0;   /*计算梯形面积 */
    printf("area=%f\n",area); /*在屏幕上显示面积的值*/
}
```

首先来看这个例子的构成。

- “/*”与“*/”之间的内容是注释。在本例中用了大量的注释，主要是提高程序的可读性，便于理解。在实际应用中，一个程序可以在一些难理解的地方



适当加上一些注释，对于程序的维护、理解有很大的帮助。注释不影响程序的执行，也不被编译。

- “#include”命令将使该程序把头文件“stdio.h”包含进来。
- main()叫做 C 程序的主函数。在 1 个完整的 C 程序中一定要有 1 个主函数 main(), C 程序总是从它开始执行。
- 一对花括号“{}”，是函数体的界限符。同时也是复合语句的界限符。
- “int up,bot,high;”和“float area;”是变量说明语句，用来通知计算机程序用到 4 个变量，并为它们分配存储单元。
- “up=3;”、“bot=5;”和“high=4;”是赋值语句，表示梯形的上底为 3、下底为 5 和高为 4。其中，“=”是赋值运算符，“3”、“4”和“5”是常量。
- “area=(up+bot)*high/2;”也是赋值语句，用来计算梯形的面积。其中，“+”、“*”和“/”是运算符，“()”是带有特定意义的运算符。
- “printf("area=%f\n",area);”是格式输出语句，“printf”是系统预定义的标准标识符，格式输出语句用来在屏幕上显示梯形的面积。
- 分号“;”表示语句结束。C 程序中要求每个语句后面要加分号。语句是 C 程序的执行单位。
- 程序中的“main”、“int”、“float”是系统规定的关键字，“up”、“bot”、“high”、“area”是用户自己定义的名字，在 C 语言中它们属于标识符。

二、C 程序的一般形式

C 程序的一般形式为：

```
#include <文件名>          /*文件包含*/
...
#include "文件名"
...
#define 宏名 宏体          /*宏定义*/
...
全局变量说明语句
main()
{
    局部变量说明语句
    执行语句
}
存储类型 数据类型 函数名 1(参数表)
{
    局部变量说明语句
    执行语句
}
...
```



存储类型 数据类型 函数名 n(参数表)

```
{  
    局部变量说明语句  
    执行语句  
}
```

从 C 程序的一般形式可以看出，C 程序是由 1 个或多个函数组成的。其中，主函数是不可缺少的，其他函数是可选的，根据需要而定。多个函数可放在 1 个文件里，也可分散放在不同的文件里。函数之间的关系是平行关系，基于此，C 语言被称为函数式语言。

函数功能的实现是由若干条语句完成的。说明语句完成数据描述，执行语句完成操作描述。语句由若干关键字加以标识，且以分号结束。

另外，C 程序中还包含预处理命令和注释。预处理命令是以“#”开头的命令。注释对程序的执行没有任何影响，编译时将被过滤掉。

三、C 程序的基本语法成份

C 程序的基本结构是函数。C 语句是组成 C 程序的基本单位，具有独立的程序功能。所有的 C 语句都以分号结尾。C 语言中有以下几种基本的语法成份：字符集、关键字、标识符、常量、操作符和分隔符。

(1) 字符集

字符是可以区分的最小符号，构成程序的原始基础。C 语言字符集是 ASCII 字符集的一部分，包括英文字母、数字以及特殊字符。

- 英文字母：a~z 和 A~Z。
- 数字：0~9。
- 特殊字符：空格、!、#、%、^、&、*、_、-、+、=、~、<>、/、\、|、.、,、:、;、?、'、"、()、[]、{}。

由字符集中的字符可以构成 C 语言进一步的语法成份，如关键字、标识符、操作符以及分隔符等。

(2) 关键字

关键字在 C 语言中有特定意义和用途，不可用于其他的字符序列。【例 1-1】中的“int”和“float”就是关键字。其中 ANSI C 标准规定的关键字有 32 个，见表 1-1。所有的关键字都用小写字母表示。在 C 语言中，大写字母和小写字母是有区别的。例如，if 是关键字，而 IF 不是关键字。

表 1-1 ANSI C 标准关键字

关键字	含义
auto	默认存储类型，意思是变量在进入代码块时被创建，离开代码块时被释放
break	无条件地退出 for、while、switch 和 do-while 语句
case	用于 switch 语句中的命令
char	C 语言中最简单的数据类型
const	防止变量被修改的修饰符



续表

关键字	含义
continue	立刻进入 for、while、do-while 循环的下次迭代
default	用于 switch 语句中，处理不与任何 case 匹配的情况
do	与 while 一起使用的循环命令，循环将至少执行一次
double	用于存储双精度实数的数据类型
else	当 if 条件为假时，用于执行另一条语句
enum	一种数据类型，这种类型的变量只能取某几个值
extern	指出变量将在其他地方被声明的限定符
float	用于存储实数的数据类型
for	循环命令，由初始化、递增和条件等 3 部分组成
goto	跳转到指定的位置
if	用于改变程序流程的命令
int	用于存储整数的数据类型
long	用于存储长整数的数据类型
register	存储限定符，指定局部变量的值存储在寄存器中
return	导致离开当前函数，并返回到调用函数的命令，可用来返回一个值
short	用于存储短整数的数据类型
signed	限定符，指出变量的值可以为正，也可以为负
sizeof	返回操作数长度（单位为字节）的运算符
static	告诉编译器应该保留变量的值
struct	用于将数据类型不同的变量组合在一起
switch	用于改变程序流程，与 case 语句一起使用
typedef	用于给变量类型和函数类型提供别名
union	让多个变量共享内存空间
unsigned	指定变量只能包含正值
void	用于指定函数不返回任何值，还可用于指定通用类型的指针
volatile	指定可以修改变量的值
while	循环语句，只要条件为真就执行一个代码段

(3) 标识符

标识符是 C 语言中用来表示变量名、数组名、函数名、指针名、结构体名和共用体名等由用户定义的字符序列。标识符是由字母、数字或下划线组成的，其中第 1 个字符必须是字母或下划线。例如，Ab、AA、A_b、_ab、aB、ab_、ab1 和 ab2 等都是正确的标识符，而 A#b、A'B、A.b、1ab、2Ab、β、λ、%r 等都是错误的标识符。

在 Turbo C 编译系统中，允许标识符最大长度为 32 个字符，有的编译系统允许标识符的有效长度为 8 个字符。



标识符不能与关键字同名，也不能与系统预定义的标准标识符（如标准函数）同名，最好能“见名知义”，以提高程序的可读性。由系统预定义的标识符称为标准标识符，由用户定义的标识符称为自定义标识符。程序设计中根据实际需要定义大量的标识符。标识符必须先定义，后使用。系统使用的一些内部标识符往往以下划线开头，为避免与系统使用的内部标识符发生冲突，用户定义的标识符一般不要用下划线开头。【例 1-1】中的“up”、“bot”、“high”和“area”就是用户定义的标识符，它们都是变量名。

(4) 常量

常量包括数值常量、字符常量以及字符串常量等。【例 1-1】中“3”、“4”和“5”就是数值常量，如'a'、'b'和'A'等都是字符常量，而“abc”、“student”和“c program.”等都是字符串常量。

(5) 操作符

操作符包括各种运算符和带有特定意义的标点符号等。【例 1-1】中的“+”、“*”和“/”就是运算符，“{”和“()”就是带有特定意义的标点符号。

(6) 分隔符

分隔符用来分隔相邻的标识符、关键字和常量。分隔符包括空格、换行符和换页符等。

了解并掌握 C 程序的结构和基本语法成份，对于 C 程序设计是非常重要的。下面举几个简单的例子。

【例1-2】 在屏幕上输出“This is a C program.”。

```
#include <stdio.h>
main()
{
    printf("This is a C program.\n");    /*在屏幕上输出*/
}
```

【例1-3】 求两个整数之和。

```
#include <stdio.h>
main()
{
    int a,b,sum;                        /*定义变量*/

    a=123;                              /*给变量赋值*/
    b=456;
    sum=a+b;                            /*求两个整数之和*/
    printf("sum=%d\n",sum);             /*在屏幕上输出*/
}
```

四、C 程序的书写格式

C 程序的书写格式完全自由，采用一定的格式主要是为了增加程序的可读性。在编写程序时应注意遵守这些约定，以养成良好的程序设计习惯。

- 为了使程序清晰易读，通常每行写一条语句，当然允许一行内写几条语句，



也允许一条语句分几行书写。

- 不同结构层次的语句从不同的位置开始，即按缩进格式书写。表示同一层次的语句行对齐，缩进同样多的字符位置。
- 花括号的书写方法较多，本书采用花括号对齐的书写方式，且左右花括号单独占一行。
- 在程序的适当地方加1个或多个空行，以增加程序的可读性。
- 适当地使用注释以增加程序的可读性，以便于维护程序。

1.4 建立与运行C程序的过程

C语言是一种编译型的程序设计语言。一个C程序从编写到得到正确的结果，一般要经过编辑、编译、连接和运行4个步骤。

(1) 编辑

编辑是用户把编写好的C语言源程序输入到计算机，并以文本文件的形式存放在磁盘上。文件名由用户自己选定。扩展名一般为“.c”，例如“a.c”、“he.c”、“h.c”等。编辑所使用的软件是编辑程序。C程序可以在Turbo C集成环境中用其编辑功能完成，也可以用其他编辑软件完成。

(2) 编译

C语言是以编译方式实现的高级语言。编译是将已编辑好的C程序文件翻译成二进制的目标代码，生成带有扩展名为“.obj”的目标代码程序。

编译前先进行预处理。在编译时，系统自动对源程序进行词法检查和语法检查。如果发现错误，会在屏幕上显示出错信息，以帮助用户修改源程序中的错误。此时，要返回到编辑步骤修改源程序，修改后，再进行编译。

(3) 连接

编译形成的目标代码程序机器可以识别，但还不能直接执行，连接是用连接程序将编译过的目标程序和程序中用到的库函数连接装配在一起，形成可执行的目标程序。可执行目标文件的扩展名一般为“.exe”或“.com”。

连接有错同样需返回到编辑步骤修改源程序，修改后，再进行编译、连接。

(4) 运行

运行是将可执行的目标文件投入运行，以获取程序的运行结果。通常在DOS操作系统提示符下直接键入可执行文件名或使用编译系统下的运行功能。

编译时产生的错误是语法错误，运行时出现的错误是逻辑错误。出现逻辑错误时需要修改算法，重新编辑、编译、运行程序。

上述4个步骤从建立到运行的全过程如图1-1所示。

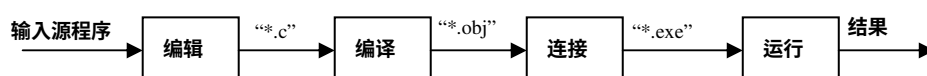


图1-1 C程序的开发过程



1.5 用 Turbo C 运行 C 程序的上机步骤

Turbo C 是美国 Borland 公司的产品, Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件, 如 Turbo BASIC、Turbo Pascal、Turbo Prolog 等, 这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品, 其中使用了新的集成开发环境, 即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。1988 年, Borland 公司又推出 Turbo C 1.5 版本, 增加了图形库和文本窗口函数库等, 而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能, 并可以在 Tiny 模式下直接生成“.COM”(数据、代码、堆栈处在同一 64KB 内存中)文件, 还可对数学协处理器(支持 8087/80287/80387 等)进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++, 它继承发展 Turbo C 2.0 的集成开发环境, 并包含了面向对象的基本思想和设计方法。1991 年为了适用 Microsoft 公司的 Windows 3.0 版本, Borland 公司又将 Turbo C++作了更新, 即 Turbo C 的新一代产品 Borland C++也已经问世了。

一、Turbo C 2.0 的安装和启动

Turbo C 2.0 的安装非常简单, 只要运行软盘或光盘的“INSTALL.EXE”即可。假设 Turbo C 2.0 安装在“E:\TURBOC2”的目录下, “E:\TURBOC2”的目录下还建立了两个子目录“LIB”和“INCLUDE”, “LIB”子目录中存放库文件, “INCLUDE”子目录中存放所有头文件。

运行 Turbo C 2.0 时, 只要运行在“E:\TURBOC2”的目录下的“TC.EXE”即可进入【Turbo C 2.0 集成开发环境】窗口, 如图 1-2 所示。



图1-2 【Turbo C 2.0 集成开发环境】窗口

二、Turbo C 2.0 集成开发环境的使用

首次进入【Turbo C 2.0 集成开发环境】窗口时, 窗口中央显示 Turbo C 的版本信息, 按任意键, 版本信息将消失, 系统处于待命状态, 如图 1-3 所示。

在【Turbo C 2.0 集成开发环境】窗口中上面一行为 Turbo C 2.0 的主菜单, 主菜单的