微 型 计 算 机(66)

C语言程序设计技巧及其应用

陈 进 编著

上海科学普及出版社

C 语言程序设计技巧及其应用

陈 进 编著

上海科学普及出版社

(沪)新登字第 305 号

unsig··

组　　稿　上海交通大学科技交流室
责任编辑　陶　洁
封面设计　毛增南

C 语言程序设计技巧及其应用
陈　进　编著
上海科学普及出版社出版
(上海曹杨路 500 号　邮政编码 200063)

新华书店上海发行所发行
上海科学普及出版社电脑照排部排版
上海市印刷七厂一分厂印刷
开本 787×1092　1/16　印张 15　字数 358000
1992 年 6 月第 1 版　1992 年 6 月第 1 次印刷
印数　1—5000

ISBN 7-5427-0575-X/TP·111　定价　平　装:11.10 元
精　装:35.00 元
(附软件)

内 容 提 要

本书由浅入深，运用了大量的实例对用C语言进行程序设计的方法和技巧作了完整的介绍。本书可分三大部分。第一部分系统、全面地描述了C语言的语法和基本特征。第二部分着重介绍了用C语言设计应用程序的经验和技巧，并对程序设计中容易出错的地方作了细致的分析。最后是应用程序举例。这些应用程序既能加深对程序设计的方法与技巧的理解和掌握，例子中的函数又可作为库函数进一步加以利用。本书具有很强的实用性，是读者掌握C语言，提高应用程序设计水平的理想的参考书。本书附软盘一张，其中包含了书中所有的函数和源程序，供读者对照运行和调用。

读者对象：程序员、大专院校有关专业师生、计算机爱好者。

前　言

近十几年我所从事的科研及教学工作大多是与计算机的应用有关的，因此，我曾先后使用过包括 ALGOL、BASIC、FORTRAN、HP-BASIC、PASCAL 以及 8086 宏汇编、68000 宏汇编等在内的多种计算机程序设计语言。这些强有力的计算及分析工具给教学及科研工作带来了极大的方便，使我受益匪浅。

尽管如此，在计算机技术迅速发展的今天，随着计算机应用水平的不断提高，即使是在一般工程应用领域中，对如何更好地、更有效地利用计算机资源也提出了越来越高的要求。其实，从工程应用的角度来看，无非是希望计算机与程序设计人员之间能有一个好的界面，希望能够比较容易而且方便地将自己的想法通过计算机加以实现。具体地说，就是要求所编制的应用程序既具有较快的计算和处理速度，同时又能够完成大量的数据处理工作，有时还需要程序具备兼顾多方面的适应能力，其中包括数据管理、通信及接口控制、图形功能等。除此之外，还希望程序设计过程尽可能地简单方便，设计的程序便于维护和更新等等。总而言之，是希望有一个良好的程序设计语言及环境。

于是，随着计算机软件知识的加深，我开始对过去只有计算机专业或专业软件设计人员才去涉足的 C 语言发生了兴趣。经过几年来的摸索和实践，再加上适用于微机系统的 C 语言本身在工程应用方面功能的加强，我感到 C 语言作为科学计算及工程应用程序的设计语言，也同样具有许多优秀之处。

在一般情况下，对于具有一定程序设计经验的人来说，要基本上掌握 C 语言，并利用它来设计一些简单的计算或分析程序以及一些小型的工程应用程序并不十分困难。但是，由于 C 语言带有一些类似于汇编语言的特征，以及其它一些独有的特性，因而，对于一般只熟悉高级语言（如 FORTRAN、BASIC 等），或初次接触 C 语言的人来说，有效地使用 C 语言、掌握其特有的程序设计方法和技巧显得尤为重要。

本书旨在将 C 语言的基本特征、一些特有的程序设计方法和技巧展现给读者，其中有不少内容是作者在使用 C 语言进行程序设计过程中所得经验的总结。希望本书中所介绍的内容能对读者的思路有一个启发，并希望 C 语言能为更多的工程应用人员所掌握，从而进一步提高我们的应用软件设计水平。

本书分为五章。前三章是 C 语言的基础，第四章介绍 C 语言程序设计技巧，第五章为应用程序举例。

在语言基础部分，本书着重描述了 C 语言的基本语法和运算符的作用。同时，在进行解释和说明的过程中，引入了大量的程序例子，有的地方还配以直观的图形解释，以加深理解。

程序设计技巧部分强调了采用结构化和模块化格式进行程序设计的方法，并对若干能提高编程效率，提高程序执行效率的方法进行了讨论与分析。此外，还分析了一些在程序设计中容易出错的问题。

最后，在应用程序举例部分，给出了若干个完整的应用程序。这些例子基本上都是从实际应用程序中挑选出来的，它们不仅可以加深读者对程序设计方法的理解，而且例子中所给

出的若干函数(子程序)本身也可以被直接引用，换句话说，相当于为读者提供了一个小的函数库。

需要指出的是本书中所给出的所有例子及其应用程序实例都是在 Microsoft C5.0 编译器下调试完成的。因此，若读者使用的是其它类型的C语言编译器，如 Lattice C、Turbo C 等，则必须对例子中与屏幕方式设置、图形等有关的函数进行适当的调整，使之与您所使用的编译器相匹配。另外，书中所给出的所有例子以及应用程序实例都已载入一张软磁盘，可供读者对照运行和调用。

学习一种计算机语言，从某种意义上讲，与学习一门外语有一定的相似之处，最好的方法便是不断地实践！只有在上机实践中才能够真正体会并牢固地掌握应用C语言进行程序设计的精髓。

本书可作为大学本科生或研究生C语言课程的教学参考书，也可供一般工程技术人员自学或参考。

作者在此谨将此书敬献给读者，希望能对您的工作或学习有所帮助。

此外，作者还要感谢上海交通大学科技交流室的罗宗信、杨福兴、蔡锡泉、徐建明四位老师，他们几位为本书的顺利出版做了大量的工作。

陈 进

91.7

目　录

第一章　C语言的基本概念 …… 1
§1.1　C语言的产生 …… 1
§1.2　C语言的特征 …… 2
§1.3　C语言程序的书写格式 …… 3
§1.4　C语言的关键字 …… 6
第二章　C语言的语法 …… 7
§2.1　程序的基本结构 …… 7
§2.2　数据的类型 …… 7
§2.2.1　基本数据类型 …… 7
§2.2.2　扩充数据类型 …… 8
§2.2.3　数据类型的重新定义 …… 10
§2.2.4　增加螺旋法则 …… 10
§2.3　运算符 …… 11
§2.3.1　赋值运算符 …… 11
§2.3.2　算术运算符 …… 12
§2.3.3　关系运算符 …… 12
§2.3.4　逻辑运算符 …… 13
§2.3.5　增量与减量运算符 …… 14
§2.3.6　位运算符 …… 15
§2.3.7　复合赋值运算符 …… 16
§2.3.8　逗号运算符 …… 18
§2.3.9　条件运算符 …… 18
§2.3.10　指针及地址运算符 …… 19
§2.3.11　Sizeof 运算符 …… 19
§2.3.12　强制类型转换运算符 …… 20
§2.3.13　运算符的优先度 …… 20
§2.4　语句 …… 22
§2.4.1　表达式语句 …… 22
§2.4.2　复合语句 …… 22
§2.4.3　if 语句 …… 23
§2.4.4　while 语句 …… 24
§2.4.5　do 语句 …… 25
§2.4.6　for 语句 …… 26
§2.4.7　switch 语句 …… 27

§2.4.8 break 语句 …… 29
§2.4.9 continue 语句 …… 30
§2.4.10 return 语句 …… 31
§2.4.11 goto 语句 …… 31
§2.5 指针与数组 …… 32
§2.5.1 指针与地址 …… 32
§2.5.2 指针与数组 …… 34
§2.5.3 指针的地址计算 …… 36
§2.5.4 指针与字符串 …… 38
§2.5.5 多维数组 …… 39
§2.5.6 指针数组 …… 41
§2.5.7 指向函数的指针 …… 43
§2.5.8 指针的类型变换 …… 45
§2.6 函数 …… 46
§2.6.1 函数的定义 …… 46
§2.6.2 函数的类型 …… 48
§2.6.3 函数调用 …… 49
§2.6.4 数组作为函数的参数 …… 52
§2.6.5 递归 …… 53
§2.6.6 main()函数的参数 …… 56
§2.7 变量的存储种类 …… 57
§2.7.1 auto 存储种类 …… 57
§2.7.2 register 存储种类 …… 58
§2.7.3 static 存储种类 …… 58
§2.7.4 extern 存储种类 …… 59
§2.7.5 静态外部变量 …… 61
§2.7.6 变量的初始化 …… 62
§2.8 结构与联合 …… 63
§2.8.1 结构的定义 …… 63
§2.8.2 结构的使用方法 …… 65
§2.8.3 指向结构的指针 …… 66
§2.8.4 结构与函数 …… 67
§2.8.5 位域 …… 69
§2.8.6 联合 …… 71
§2.9 预处理程序 …… 73
§2.9.1 宏定义(#define) …… 74
§2.9.2 文件插入(#include) …… 74
§2.9.3 条件编译(#if、#ifdef、#ifndef、#else、#endif) …… 75
第三章 C 语言的标准库函数 …… 79

§3.1 标准库函数…… 79
§3.2 标准输入输出函数…… 80
§3.2.1 字符输入输出…… 80
§3.2.2 格式化输入输出…… 81
§3.3 文件输入输出函数…… 86
§3.3.1 流式文件输入输出函数…… 86
§3.3.2 流式文件的打开与关闭…… 87
§3.3.3 流式文件的定位…… 88
§3.3.4 流式文件的读写…… 89
§3.3.5 低级文件输入输出函数…… 93
§3.4 字符串处理函数…… 95
§3.4.1 字符串长度…… 96
§3.4.2 字符串复制…… 96
§3.4.3 字符串比较…… 97
§3.4.4 字符串连接…… 97
§3.4.5 字符串初始化…… 98
§3.4.6 字符串中的字符查找…… 99
§3.4.7 字符串转换…… 99
§3.5 内存管理函数 …… 100
§3.5.1 内存分配 …… 100
§3.5.2 内存释放 …… 102
§3.6 数学库函数 …… 102
第四章 C语言的程序设计技巧…… 105
§4.1 C语言程序的习惯写法及规律 …… 105
§4.1.1 C语言程序的设计过程 …… 105
§4.1.2 C语言程序的书写规律 …… 105
§4.2 程序设计的一些基本方法 …… 109
§4.2.1 无限循环的构造方法 …… 109
§4.2.2 结束从标准输入设备(键盘)输入数据的方法 …… 110
§4.2.3 文件输入输出时的附加处理 …… 111
§4.2.4 使用指针读入字符串的方法 …… 112
§4.2.5 直接读写内存的方法 …… 112
§4.2.6 使用指针定义输出格式 …… 113
§4.3 结构化的程序设计方法 …… 113
§4.3.1 使用合理的程序书写格式 …… 113
§4.3.2 使用规范化的控制结构 …… 114
§4.3.3 注意程序的模块化 …… 116
§4.3.4 灵活运用预处理程序 …… 117
§4.4 提高程序执行效率的方法 …… 118

§4.4.1 运算符的使用 …… 118
§4.4.2 条件式的安排 …… 119
§4.4.3 利用指针处理多维数组 …… 119
§4.4.4 函数与宏定义 …… 119
§4.4.5 低级文件输入输出与流式文件输入输出 …… 120
§4.4.6 合理使用中间变量 …… 122
§4.5 程序设计中容易出错的地方 …… 123
§4.5.1 =与== …… 123
§4.5.2 优先顺序 …… 124
§4.5.3 非法表达式 …… 124
§4.5.4 数组的下标范围 …… 125
§4.5.5 初始化方法 …… 125
§4.5.6 结构变量的赋值方式 …… 126
§4.5.7 函数的参数 …… 127
§4.5.8 其它 …… 132
§4.6 C语言程序与汇编语言程序的连接 …… 133
§4.6.1 用汇编语言为C语言编制函数 …… 133
§4.6.2 用于被Lattice C调用的汇编函数 …… 134
§4.6.3 用于被MS-C调用的汇编函数 …… 135
§4.6.4 C语言程序对汇编过程的调用 …… 136
第五章 应用程序举例 …… 137
§5.1 三次样条插值 …… 137
§5.1.1 边界条件 …… 138
§5.1.2 约束边界条件 …… 139
§5.1.3 自由边界条件 …… 144
§5.1.4 周期边界条件 …… 146
§5.2 快速逆拉普拉斯变换(FILT) …… 153
§5.3 快速傅里叶变换(FFT) …… 163
§5.3.1 基2的FFT …… 163
§5.3.2 时间抽取型FFT …… 167
§5.3.3 频率抽取型FFT …… 169
§5.3.4 基4的FFT …… 171
§5.3.5 实数数据的FFT …… 173
§5.3.6 FFT程序的检验 …… 177
§5.4 工具型应用程序 …… 182
§5.4.1 通用函数模块 …… 182
§5.4.2 简易全屏幕编辑器 …… 187
§5.4.3 计算机之间的文件传送 …… 203
§5.4.4 磁盘信息查找 …… 214

§5.4.5 源程序分页打印输出 …… 222
参考文献 …… 226

第一章　C 语言的基本概念

§1.1　C 语言的产生

C 语言是一种通用的计算机程序设计语言，它既可用来编写系统程序，又可以用来设计一般的应用软件。C 语言与 UNIX 操作系统有着十分密切的关系。这主要是由于 C 语言是基于 UNIX 操作系统之上发展形成的。另一方面，UNIX 操作系统以及运行其上的绝大部分软件又都是用 C 语言编写而成。

UNIX 的最初版本是 1970 年由贝尔实验室(Bell Laboratories)的 K. Thompson 和 D. M. Ritchie 等完成的，当时的 UNIX 主要是作为 PDP-7 型计算机的操作系统，并且全部用汇编语言写成。K. Thompson 和 D. M. Ritchie 在开发 UNIX 系统的同时，还对 M. Richards 所提出的 BCPL 语言作了改进，形成了一种被称为 B 的语言，然后在 B 语言的基础上又进行了进一步的完善，最终形成了 C 语言。之所以取名为 C，就是因为它派生于 B 的缘故。图 1.1 展示了 C 语言及其它几种常用语言的产生过程。显然，C 语言属于 ALGOL 语言体系。

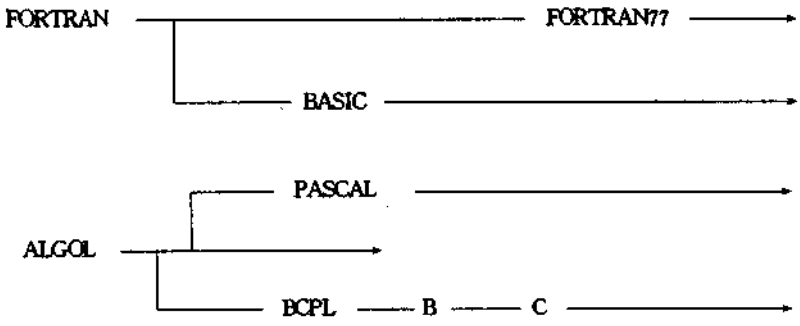

图 1.1　几种程序设计语言的历史

C 语言大约形成于 1972 年。1973 年 D. M. Ritchie 把原来用汇编语言写成的 UNIX 系统中 90%的部分改由 C 语言重写，并在 PDP-11 型小型计算机上用 C 语言及少量汇编语言完成了沿用至今的 UNIX 操作系统。UNIX 系统改用 C 语言写成后，既有利于 UNIX 系统向其它不同类型计算机的移植，使各种类型的计算机都能够使用 UNIX 操作系统，同时，又促进了 C 语言的不断发展和普及。

尽管 C 语言是与 UNIX 系统同时发展起来的，但如今它已经是一种不单是能在 UNIX 环境下工作的语言了。实际上，C 语言已经发展成为能在多种机型、多种操作系统，尤其是能在微型计算机及其操作系统的支持下进行程序设计的一种功能极强的通用计算机语言。而且，仅对微型计算机而言，C 语言的版本已有多种。目前在我国较为流行的微机 C 语言版本有在 MS-DOS 操作系统支持下的 Lattice C，Turbo C 和 Microsoft C。表 1.1 归纳了近年来专门为微型计算机而设计的若干种 C 语言编译器。

表 1.1　用于微型计算机上的 C 编译器

操作系统(OS)	C 编译器名称
CP/M	BDS C Lattice C Whitesmith C AZTEC C I DR-C
MS-DOS	OPTIMIZNG C86 Desmet C Lattice C Whitesmith C HI-TECH C Turbo C MS C(Microsoft C)
OS-9	MICROWARE C

§1.2　C 语言的特征

C 究竟是一种什么样的语言呢？让我们首先根据图 1.2 所给出的 BASIC、PASCAL 和 C 三者之间的关系来进行一些分析。

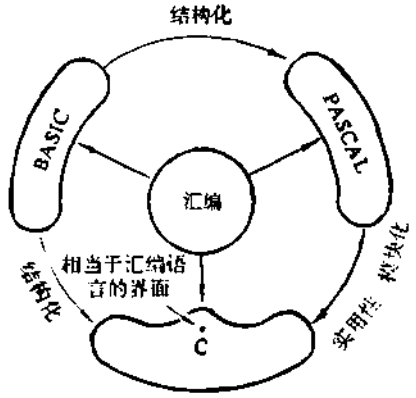

图 1.2　C 语言的特征

首先，图的中央部分为与硬件有着密切关系的汇编语言，从理论上讲只要具有汇编语言，就可以完成所有的软件设计工作，但由于这种语言是专门针对硬件而不是面向人设计的，因此，必然具有效率低的缺点。为此，人们开始考虑针对某些方面的工作设计，并便于掌握和理解的语言，最初出现了适用于科学及工程计算领域的 FORTRAN 语言，商业事务处理方面专用的 COBOL 语言等。

接着，我们再来看看能在微机上运行的 BASIC、PASCAL 和 C 语言，它们所具有的一个共同点就是这三种语言均是相对于汇编语言为提高软件的生产性而发展起来的。

BASIC 语言本是为学生的教学培训而从 FORTRAN 中派生出来的。随着微机的普及，BASIC 语言也达到了惊人的普及程度，这主要是因为一开始 BASIC 语言就作为微机的标准语言而随机配备，再加上它采用解释执行方式，运行、纠错比较简单，语言自身也并不十分严格，程序格式也极为自由，尤其是对于初次接触计算机的人来说非常容易接受。

PASCAL 和 C 语言均是在 ALGOL 的基础上发展起来的，两者具有许多相似之处。如果要说两者有什么不同的话，那就是 PASCAL 是面向教育而设计的，而 C 语言则是实实在在地着眼于实际应用。与 BASIC 相比，C 和 PASCAL 都十分强调程序的结构化、简单化和易读性。C 语言还进一步强调程序的模块化，这就意味着可以将程序分割成若干部分独立、并行地进行编制。

C 语言是一种相对比较"低级"的语言，它具备一些过去只有汇编语言才能够实现的功能。起初，C 语言主要是用来替代汇编语言编写操作系统、编译器、数据库以及机械控制程序等。随着它自身的不断强化和完善，更由于它具有处理速度快、操作方便等优点，今天已经在科学及工程应用领域中受到了普遍的重视和欢迎，并且在这些领域中发挥了积极的作用。归纳起来，C 语言主要具有如下特征：

(1) C 语言是一个比较小的语言体系，是现有程序设计语言中规模最小的语言之一。小的语言体系往往能够设计出较好的程序，C 语言的关键字很少，标准情况下只有 29 个。

(2) C 语言非常容易移植。这也是因为 C 语言本身比较小，并且一开始就产生于小型计算机。同时还由于 C 语言本身没有输入输出功能，输入输出是通过库函数来实现的。

(3) C 语言很简洁，而且重视实用性。C 语言具有众多的运算符，使用方法也比较独特，如位操作，移位操作等；多个语句(如间接地址计算等)还可以并入一条语句或一个表达式中。

(4) C 语言具有由函数的集合所构成的模块化结构。这样，编程者就可将一个大型程序分割成若干部分并分别由不同的人员同时编写。因此，C 语言也是一种具有结构化和模块化特性的语言。

(5) C 语言中可以使用指针，这就使它具备了其它高级语言都没有的近似于汇编语言的功能。

(6) C 语言具有最新的流程控制结构，从而保证了程序的结构化、简易化并易于更新、维护。

(7) C 语言具有极强的数据定义方法，几乎可以定义所有类型的数据。并且，数据类型之间的相互转换也十分自由。

(8) C 语言允许使用递归函数结构，还具有宏定义功能。这一特性极大地丰富了 C 语言的表达能力。

(9) C 语言允许用户将自己定义的新命令加入语言之中。实质上，C 语言的程序完全是由模块化的函数集合所组成。用户对函数的定义本身就可以看作是对语言的一种扩充。

当然，C 语言并不是完美无缺的。例如，C 语言没有自动检查数组边界的功能，有些符号如 * 和 = 具有多重用途等等。因此，在使用中往往容易引起一些不必要的错误，稍不注意就可能导致意想不到的结果。

§1.3 C 语言程序的书写格式

学习一种新的程序设计语言的唯一有效方法就是经常编写一些程序。那么，编写程序应该如何开始呢？在这里，首先就 C 语言程序的一般形式进行一些说明。其实，C 语言对程序的书写格式几乎没有任何限制，可以采用完全自由的形式，对语句位置的唯一要求就是预处

理命令必须放在程序的最前面。尽管如此，为使程序易于理解、便于阅读，在编程中还是应该考虑遵循一定格式的。

下面我们通过一个例子来看看应该如何书写 C 语言程序。

例 1.1　打印输出“C programming language ”的程序

```
#include <stdio.h>                          /*line1*/
                                            /*line2*/
main()                                      /*line3*/
{                                           /*line4*/
    printf("C programming language");       /*line5*/
}                                           /*line6*/
```

(1)预处理命令

通常，程序的开头是预处理命令，如第一行的 #include<stdio.h>。stdio.h 一般被称为“头文件”，它包含有程序在编译时的必要信息。该预处理命令行通知编译器在编译时先读入 stdio.h 文件一起参加编译。一般 C 编译系统都会提供若干个不同用途的头文件。

(2)main()函数

程序的正文，由 main()函数开始，这是一个特殊的函数(注意“()”不能遗漏)。C 语言规定程序的执行从 main()函数开始，函数体由大括号“{”和“}”括起来，函数中每条语句结束时都要加上分号“;”进行标识。

(3)注释

在程序中加入适当的注释会有助于对程序的阅读理解和调整。C 的注释由/* 开始，然后由 */结束，具有如下所示的形式：

```
/*    注释    */
```

当然也可以写成多行，如：

```
/*
      注释
*/
```

由于在编译时编译器会自动跳过注释部分，因此注释可以放在程序中的任何位置。

(4)大写字符与小写字符

在 C 语言的程序中既可以使用大写字符，也可以使用小写字符。一般的习惯是在普通情况下采用小写字符，而对一些具有特殊意义的变量或常数则采用大写字符。但是，必须记住的是，同一字符的大写和小写是有区别的，即在 C 语言中同一字符的大写和小写会被认为是两个不同的字符，例如：

Abc 和 abc

在编译时将被解释成两个不同的变量。

(5)编程格式

前面已经介绍，C 语言程序的书写方式十分自由，也就是说，可以采用任意的方式编写程序。但是，如果在编程时不注意程序的书写格式，即使是本人，恐怕在过了一段时间之后也难以明白原来程序的内容，更不用说要让他人来阅读或修改程序了。

为了使程序便于理解，设计时应注意这样几个问题：

①使用模块化结构，将不同的处理功能分别作成不同的函数。

②一个函数的处理内容不宜过多，应保持函数的“小型化”。

③在一个函数内部，按功能块分层次“缩格”书写。

有关函数将在后面章节进行专门论述，下面只给出一个按功能块分层次“缩格”书写的程序例子。

例 1.2　按功能块层次“缩络”编写程序举例

```
#include <stdio.h>
main()
{
    int temp, i, j;
    int a[10], b[10];

    for( i=0; i<10; ++i )
    {
        scanf("%d",&a[i]);
        b[i] = a[i];
    }
    for( i=0; i<10; ++i )
        for(j=i+1; j<10; ++j )
            if( b[i] < b[j] )
            {
                temp = b[i];
                b[i] = b[j];
                b[j] = temp;
            }
        for( i=0; i<10; ++i )
        Printf("%5d %5d\n",a[i],b[i]);
}
```

注意：习惯上一次缩格选取 8 个或 4 个字符位置，可用 Tab 键进行控制。

(6)插入文件

C 语言程序被编译时，编译器可从其它文件中读入程序或数据一起参加编译，这是 C 语言的一个非常有价值的概念，根据这一事实，C 语程序可以被划分成若干个小的部分，以便于编辑、管理、修改和维护。此外，某些常用的程序在一次设计完成之后，可以供不同的 C 语言程序反复调用，这样保证了在多个不同的程序中能够部分地利用同一程序或者数据定义而不致于出错。

文件读入是通过预处理命令

#include <file name>　或　#include "file name"

在对程序进行编译时完成的，被读入的文件只参加编译。

(7)转义序列

对于不可显示的控制字符，可以采用以反斜杠“\”开头的专用转义字符来表示。利用转义序列可以完成一些特殊功能和打印输出时的格式控制，常用的转义序列如表 1.2 所示。

表 1.2 常用转义序列

转义序列	含　义	ASCII 码(16 进制)
\n	换行(LF)	0x0d
\r	回车(CR)	0x0a
\t	水平制表(HT)	0x09
\b	退格(BS)	0x08
\f	换页(FF)	0x0c
\'	单引号	0x27
\\	反斜杠	0x5c
\0	空字符(NULL)	0x00
\ddd	任意一文字的 8 进制码,ddd 为一个 3 位 8 进制数	

§1.4 C 语言的关键字

关键字有时也被称为保留字,它们是指在 C 语言中被赋予了严格定义的独特标识符。在程序文本中,关键字不允许被重新定义或作其它用途使用。表 1.3 按类别列出了 C 语言的关键字。

表 1.3 C 语言的关键字

存贮种类	数据类型补充	数据类型	语　句	其　它
auto	long	char	break	sizeof
extern	short	double	case	typedef
register	unsigned	enum	continue	void
static		float	default	
		int	do	
		struct	else	
		union	for	
			goto	
			if	
			return	
			switch	
			while	

注:①在新近的一些 C 语言版本中已将下列标识符列入关键字范围:
ada、asm、fortran、pascal。
②关键字务必全部使用小写字符。

与其它比较常用的语言相比,C 语言的关键字是很少的,例如 Ada 语言就有 62 个关键字。注重灵活运用各种关系的搭配组合,而不是设定更多的特殊符号和关键字,这正是 C 语言的一个重要特征。