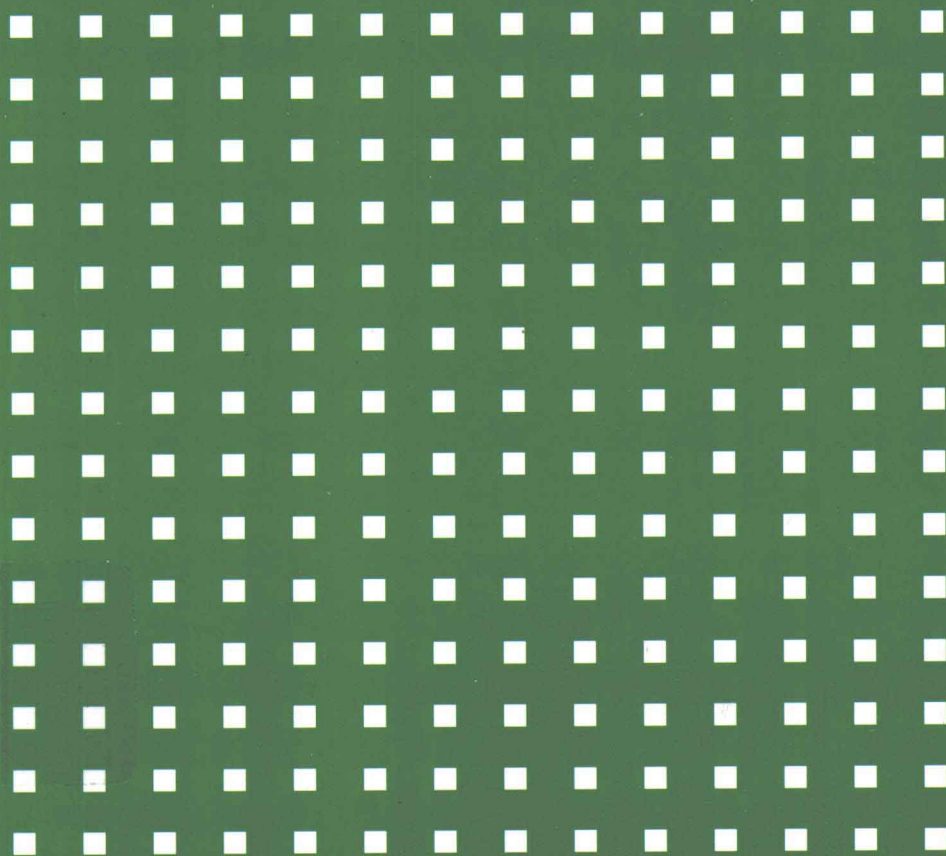


高等学校计算机专业教材精选·算法与程序设计

普通高等学校优秀教材二等奖

PC C语言教程 (修订版)

李文兵 等编著



清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

PC C语言教程 (修订版)

李文兵 等编著

清华大学出版社
北京

内 容 简 介

本书是介绍 C 语言及其程序设计技术的教科书,由 11 章组成,即第 1 章 绪论,第 2 章 基本数据类型及数组,第 3 章 输入输出与库函数,第 4 章 变量的存储类别,第 5 章 运算表达式,第 6 章 C 程序结构,第 7 章 宏指令的用法,第 8 章 指针的用法,第 9 章 结构数据设计,第 10 章 函数的调用方法,第 11 章 文件处理程序设计。这样的章节安排,便于在介绍 C 语言及其程序设计的同时,探讨方方面面的 C 程序设计技术。

本书注重 C 语言的低级性和系统描述功能,并力求做到突出 C 语言的专业性与教材通俗易懂的统一。本书系统性好,结构合理,可读性好,适合作 C 程序设计课程的教材,也可作为 C 程序设计者的参考用书。

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

PC C 语言教程/李文兵等编著. —修订版. —北京:清华大学出版社,2012.1

(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-27078-2

I. ①P… II. ①李… 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 207589 号

责任编辑:汪汉友

责任校对:李建庄

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:保定市中国画美凯印刷有限公司

经 销:全国新华书店

开 本:185×260

印 张:20.5

字 数:512 千字

版 次:2012 年 1 月第 1 版

印 次:2012 年 1 月第 1 次印刷

印 数:1~3000

定 价:33.00 元

产品编号:040450-01

出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析问题、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,我们相信它能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前 言

本书是作者从事 C 语言教学和软件开发的体会与经验总结,现在与读者见面了,欢迎读者提出宝贵意见和建议。

1988 年 4 月,作者在国内出版了第一本可在个人计算机上进行编辑、编译和执行的 C 语言的教科书《IBM PC C 语言简明教程》,并于 1990 年 2 月又在清华大学出版社出版了个人计算机使用的《C 编译系统》集成软件。此外,作者还陆续出版了《IBM PC C 语言例题 习题 库函数》、《C 语言学习指导》、《Turbo C 及其应用》、《PC C 语言教程》和《C 应用程序设计技术》。其中,《PC C 语言教程》荣获普通高等学校优秀科技教材二等奖,本书是该书的修订版。

在这些著作中,作者提倡学习者掌握 C 的低级性和系统描述功能,不断提高独立自主开发函数的能力。作者认为,只有这样,我国的 C 语言教学才能为培养系统软件大师打下语言基础。

本书就是基于这种想法而编著的,具有如下特点。

(1) 突出 C 语言的低级性和系统描述功能。C 语言是为开发 UNIX 操作系统而诞生的系统描述语言,是迄今为止开发系统软件的最好语言。目前在高校相关专业的教学中摒弃了一些其他语言,而把 C 语言作为重要的专业基础课和工具课。之所以如此,就在于 C 语言的低级性和较强的系统描述功能。因此,学习 C 语言,就要掌握它的这些特点与功能。只有掌握了 C 语言的低级性和系统描述功能,才能善于编写控制硬件的程序、与系统软件链接的程序、与汇编程序混编的程序,从而提高编写系统软件的能力及软件的质量。本书在数据类型及其转换、输入输出、库函数、内存的数据存储及读写、宏指令使用、指针应用、字段结构、文件处理和行命令等内容上,都注重了 C 语言的低级性和系统描述功能的介绍。

(2) 突出 C 语言的专业性和教材通俗易懂的统一。C 语言程序设计是专业性很强的一门课程,它与计算机硬件、汇编语言关系密切。因此,一些学员感到 C 语言程序设计难学难用。为了使本书既介绍出 C 语言的专业性较强的内容,又能使本书通俗易懂,作者运用了如下方法。

① 语法现象用实例程序来说明 全书共有 264 个练习程序用来说明各种语法问题。这样就使语法问题变得很容易理解,且能学而致用。

② 涉及有关硬件、复杂结构数据的问题,使用结构图来说明,问题便一目了然。

③ 对难点问题进行详细说明。

④ 文字言简意赅,深入浅出。

这些方法的运用,使本书通俗易懂,C 语言的一些重点和难点问题,读者在使用本书时会迎刃而解。

(3) 本书系统性好,结构合理。本书的结构既便于介绍 C 语言及其程序设计,又便于介绍 C 语言程序设计技巧及注意问题。

(4) 本书根据美国标准(ANSI)编著,程序全部用 Visual C++ 6.0 编译执行通过。

在本书的编著过程中,王玉华、黄硕之、李海迎、贾雯、王颖、郝玲、刘畅、李鸿桐、李洪等老师参编了部分章节及调试程序。

感谢清华大学出版社对作者的一贯支持。

感谢广大读者!

李文兵

2011 年 12 月

目 录

第 1 章 绪论	1
1.1 C 语言	1
1.2 C 程序	3
1.3 上机操作	8
思考题	16
第 2 章 基本数据类型及数组	17
2.1 整型数据	17
2.2 字符型数据	21
2.3 浮点数	27
2.4 类型转换	31
2.5 数组	38
第 3 章 输入输出与库函数	48
3.1 字符的输入输出与处理函数	49
3.2 按格式输出函数	52
3.3 按格式输入函数	57
3.4 字符串处理函数	64
第 4 章 变量的存储类别	70
4.1 局部变量与全局变量	70
4.2 自动存储变量	73
4.3 外部存储变量	77
4.4 静态存储变量	81
4.5 寄存器存储变量	85
第 5 章 运算表达式	88
5.1 表达式	88
5.2 算术运算表达式	92
5.3 位逻辑运算表达式	95
5.4 赋值运算表达式	98
5.5 关系运算表达式	100
5.6 递增和递减运算表达式	102

5.7	基于指针的运算表达式	105
5.8	逻辑运算表达式	108
5.9	其他运算表达式	115
第 6 章	C 程序结构	119
6.1	条件结构与分支结构	119
6.2	循环结构	126
6.3	嵌套结构	128
6.4	中断结构	131
6.5	复合程序结构	137
6.6	程序结构设计技巧	140
第 7 章	宏指令的用法	160
7.1	宏定义	160
7.2	带参数的宏	166
7.3	条件编译结构	173
7.4	嵌入文件宏指令	179
7.5	编译源文件用的宏名和宏指令及宏体参数操作符	183
第 8 章	指针的用法	187
8.1	指针的运算	187
8.2	指针参数	191
8.3	内存空间的动态分配	197
8.4	多级指针	200
8.5	指针数组	205
8.6	指针的指针数组	209
8.7	数组指针	213
第 9 章	结构数据的设计	217
9.1	结构	217
9.2	结构数组	224
9.3	结构指针	228
9.4	结构嵌套	236
9.5	结构指针数组	238
9.6	返回结构数据的函数设计	241
9.7	共用体	249
9.8	枚举类型数据	256
9.9	字段结构	263

第 10 章 函数的调用方法	267
10.1 函数调用的基本方法.....	267
10.2 函数的嵌套调用与递归调用.....	271
10.3 使用函数指针的函数调用.....	278
10.4 外部调用.....	287
10.5 函数调用过程的参数传递方式及其参数设置.....	289
 第 11 章 文件处理程序设计	 297
11.1 文件的概念.....	297
11.2 流式文件的输入输出.....	299
11.3 文件输入输出的重定向.....	304
11.4 文件的随机读写.....	308
 参考文献.....	 316

第 1 章 绪 论

1.1 C 语 言

1. C 语言产生的历史背景

C 语言是 20 世纪 70 年代贝尔(Bell)实验室为描述 UNIX 操作系统和 C 编译程序,而开发的一种系统描述语言。

C 语言是美国贝尔实验室的研究人员 Ken Thompson 和 Dennis M. Ritchie 合作研究开发出来的。他们的研究历程如表 1.1 所示。

表 1.1 C 语言的产生过程

年 份	研 究 人 员	研 究 成 果
1969—1970	Ken Thompson,Dennis M. Ritchie	在 PDP 机上用汇编语言编写出 UNIX
1970—1971	Ken Thompson	在 BCPL 语言基础上开发出 B 语言
1971—1972	Dennis M. Ritchie	利用 B 语言开发出 C 语言
1973—1975	Ken Thompson,Dennis M. Ritchie	用 C 编写 UNIX 第 1~6 版
1978	W. Kernighan,Dennis M. Ritchie	根据 UNIX 第 7 版,出版了《The c programming Language》一书

1978 年以后,C 语言便被移植到大多数计算机上,小至 8 位微型计算机,大至像 Cray 1 这样巨型计算机。目前 C 语言已运行在各种操作系统和各种机种上。

2. C 语言的特点

对于 C 语言的特点,作者认为有如下 5 点。

(1) 系统的小型化 C 语言作为一种标准语言得到广泛应用,原因就在于 C 语言能够被移植到各种操作系统上。C 语言之所以移植性好,原因是其系统很小,很紧凑。C 语言没有定义输入输出,而且关键字用得少。在语法上仅仅定义了基本的控制语句;而输入、输出和字符处理等,是用外部函数,连接到所编写的程序上实现的。这样,C 语言就能够适用多种多样的环境。用 C 语言所编写的程序在这些环境之间进行移植时,只需注意如下两个问题即可。

① 各变量类型及其字长问题。各变量类型的字长因所用的 CPU 的字长而异,故移植程序时必须注意。

② 与操作系统接口有关的库函数问题。把 C 语言从一种操作系统移植到另一种操作系统上时,必须修改有关的库函数。因为这些库函数是在特定环境下定义的,环境一改变,它就不适应了。美国标准(ANSI)之所以规定 C 编译系统必须提供标准库函数集合,其目的就是为了增强 C 程序的可移植性。

(2) 语言的低级化 用 C 语言编写的程序中最大且最有名气的要算 UNIX 了。这样大

的操作系统软件之所以使用 C 语言编写,就在于 C 语言的低级性。语言越低级,硬件控制能力就越强。由于 C 语言的类型检验能力比较弱,能直接访问物理地址等特点,使得它在直接控制硬件方面,能实现同汇编语言相同的效果。这就是能用汇编语言所编写的操作系统可以使用 C 语言来编写的原因。

由此看来,实际上 C 语言所能迅速应用到的领域正是以前汇编语言所应用的领域。因此,C 语言被称作是“便携汇编语言”。它是介于高级语言和汇编语言之间的一种语言,也正因为如此,有些专家便称 C 语言为中级语言。

(3) 应用的双向化 事实上,由于 C 语言有较强的绘图能力,能处理汉字,能直接管理硬件,能链接 BIOS、OS 和高级语言,能与汇编语言混合编程,故用它来设计管理信息系统,方法会更灵活,功能会更强,效果会更加完美。因此,可以说,C 语言既具有高级语言的特点,又有汇编语言的特点,它既是成功的系统描述语言,又是一种实用的程序设计语言。这就是所谓的 C 语言应用的双向化。

(4) 程序编写的自由化 C 语言程序设计在语言自身的规格上没有什么严格的规定,可以相当自由地进行编写。一行中的语句之间可以有空格,也可以无空格。行与行之间可以空行,也可以无空行。可以说 C 程序的书写格式是相当的自由。为便于阅读,C 程序习惯上设计成锯齿状。

对于程序员来说,若使用高级语言,只要按照语法规则来编写,就能够比较顺当地编写出正确的程序,并能自动地进行错误检验。而 C 语言,由于其自由度较大,对变量类型、变量所表示的范围和存储空间存取的制约就少。这样,错误检验在各个阶段难以进行,要用它编写出正确的程序是有那么一点点难度。这意味着,同其他语言相比,C 语言具有面向专业的特征。

(5) 表示方法的简捷化 这表现在如下几个方面。

① 语句简洁 这一点从 C 语言与 Pascal 语言的比较就可以看出,如表 1.2 所示。

表 1.2 C 与 Pascal 的语句比较

C 语言	Pascal
{ }	begin end
if(e)s;	if e then s
int i;	var i;integer;
int f();	function f():integer;
int a[];	var a::array[...]of integer
int * p;	p: ↑ integer

② 变量更新表示简单 在 C 语言中

i=i+1;

可以表示成

i+=1;

③ 变量的自增/自减表示简单

i++; i--;

(1)

++i; --i;

(2)

注意：式(1)和式(2)的相同点是运算后*i*都是自动加/减1；式(1)和式(2)的区别是++*i*/--*i*是先进行加/减1再使用，而*i*++/*i*--则是先使用再进行加/减1。

④ 关键字少 ANSI 只定义了 32 个关键字，即

auto	double	int	struct
break	else	long	switch
case	enum	register	typeder
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

注意：这些关键字都是用小写字母定义的。使用时，不能用大写字母代替；不能挪作他用，不允许用它们作变量名、函数名。

⑤ 语句少 C 语言只定义了如下 10 个流程控制语句。

if	do...while	switch	continue	goto
while	for	break	return	;(空语句)

1.2 C 程 序

1. C 程序的格式

用 C 语言所编写的程序，称作 C 语言源程序，简称 C 程序。C 程序一般由一个或多个函数组成，这些函数可集中放在一个或分散在几个文件中，为表示这些文件都是 C 程序文件，各文件名都必须以 .c 结尾，如 expl_1.c 就是一个 C 程序文件名。该文件只含有一个函数 main()，如练习 1.1 所示。

【练习 1.1】

```
expl_1.c
行号
1  #include <stdio.h>
2  main() /* ...add of a and b ... */
3  {
4      int a,b,sum;
5      a=123;b=456;
6      sum=a+b;
7      printf("sum is %d\n",sum);
8  }
```

就该练习，对 C 程序，说明如下：

① C 程序无行号，这里的行号是为了便于讲解而加的。

② expl_1.c 是文件名。在这里，读者看到 C 程序文件名的结尾是 .c。c 即表示 C 源

文件。

③ `/*` 与 `*/` 之间的内容是注释。本例中,它说明 `main()` 函数是求解 `a` 和 `b` 两个自变量的值之和。注释不被编译,只起备忘和帮助阅读的作用。

④ 在 C 程序中,其后紧跟括号 `(( ))` 的标识符是函数名。已经看到,该程序只有一个函数,即函数 `main()`。`main` 是表示程序的主体部分,是主程序的约定词。`main()` 可称为主函数。

在执行 C 程序时,一调用主函数 `main()`,程序就开始执行;而 `main()` 返回时,程序执行便终止。

在一个完整的 C 程序中一定要有一个主函数 `main()`,不管这个函数放在什么地方,C 程序总是从它开始执行。函数 `main()` 可以调用其他函数;其他函数之间也可以有调用和被调用关系;此外,一个函数还可以调用其本身,这称为递归调用。

⑤ 花括号 `{ }` 是函数体的界限符。

⑥ 第 4 行为说明语句。说明变量 `a`、`b` 和 `sum` 的值为整型 (`int`) 数。该说明语句说明:

- 该程序使用了 `a`、`b` 和 `sum` 这 3 个变量;
- 这些变量的值是整数。

这个程序里出现的变量 `a`、`b` 和 `sum` 的含义,正如图 1.1 所示,是给计算机内存单元起的名字,各单元内存放着各自数据,分别为 123、456、789。

注意: 这 3 个变量名所表示的 3 个内存单元未必是相邻的。之所以称为变量,是因为重新给它们赋值后,它们的值发生了变化,故把它们叫做变量。

例如:

`a=234;b=567;sum=a+b;`

执行后,`a`、`b`、`sum` 的内容便成为如图 1.1 (b) 所示的情况。

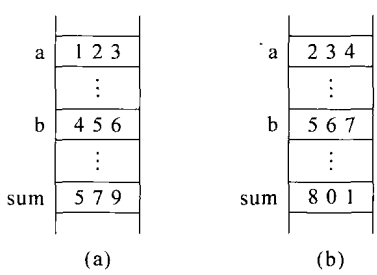


图 1.1 变量的含义

⑦ 第 5、6 行中出现的 `=`,是赋值运算符,表示把赋值运算符右边的数值或是运算结果赋值到其左边的变量所指定的存储单元中。在 C 语言中赋值运算符左边的操作数叫左值,右边的叫右值。

⑧ 第 7 行中的 `printf()` 是按格式输出函数。它是编译系统中编好的函数。由于这类函数是编译系统中带来的,故俗称库函数。这里先简单介绍一下这个库函数,以便使用。

`%d` 是 `printf()` 函数的最简单的打印格式,叫转换控制。它指出把后面的相应参数值 (这里是 `sum` 的值),以所需要的最少位,按十进制打印,前后不留空格。

⑨ 语句后面要加分号 `;`,表示语句结束。带有分号 `;` 的语句叫简单语句,它是执行单位。在 C 程序中 `printf()` 语句和赋值语句是很常用的语句。

在 C 程序里,不是每个简单语句必须占据一行。一行中可以有若干个语句,如练习 1.1 中第 5 行所示。但每个简单语句后,必须要有一个分号 `;`。

有一点要特别注意,即花括号 `{ }` 前一个语句末的分号不能少,而在 `}` 后面不能有分号。

⑩ 第 1 行中的 `#include <stdio.h>` 是一条宏指令。它的功能是把文件 `stdio.h` 包含到该程序中,因为函数 `printf()` 是在 `stdio.h` 文件中定义。只有这样,该程序才能正确使用

函数 printf()。

⑪ 该程序的执行结果是

```
sum is 579
```

对于 C 程序必须强调如下几点。

(1) 在 C 程序里提倡把一个程序分割为若干个函数来编写,又提倡把各个函数分别放在不同的文件里。这主要是考虑到一个有实用价值的程序往往需要反复进行修改、调试。如果把程序编制成一个大程序,尽管有时看起来还算紧凑、简洁,但只要修改一处,也要重新编译一次整个程序,这在时间上会造成很大的浪费。如果把一个程序分成若干个函数来编写,并把这些函数分别放在不同的文件里,那么修改某个函数,就只需要重新编译一下这个函数所在的文件,而其他未修改的函数所在文件就无须重新编译,这样编译时间会大大减少。待各个文件的编译都通过之后,通过文件链接或函数调用等方法,就可以把这些函数链接成一个完整的程序来执行。

(2) 程序中函数彼此间的排列顺序如何是没关系的,当然尽可能整理成便于理解整个程序的结构顺序为好。

(3) C 语言不允许在函数的定义中再定义函数,例如下面这段程序就是非法的,错误的。

```
f()
{
    :
    g()
    {
        :
    }
    :
}
```

从这个意义上可以说,C 程序中的函数都是一个级别的。

2. 最小 C 程序

最小 C 程序如下所示:

```
mop() { }
```

显然,mop()是个函数名。在这个程序里,因为{ }内没有任何可执行的语句,所以该函数一旦被调用,什么也不做就立即返回到调用它的函数,虽然如此,这个函数还是很有价值的。在程序开发时,可以使用这个最小程序,为将来要设置的函数事先安排一个位置。

这个最小 C 程序,实际就是一个最小函数。

3. 函数的定义形式

一般函数的定义以如下形式给出:

```
函数属性说明  函数类型说明  函数名(参数表)
参数类型说明
{
```

```

    变量定义部分
    执行语句部分
}

```

函数定义中必不可少的部分是

```

函数名 ( )
{
}

```

其他部分根据需要来确定有无。

函数定义可以分为两部分来说明：函数说明部分和函数体。

(1) 函数说明部分 函数说明部分包括如下几部分。

① 函数属性说明 该部分用于指出函数是内部函数，还是外部函数；分别用关键字 `static` 和 `extern` 表示。内部函数只能被该函数所在文件内的函数调用；而外部函数则可以由其他文件的函数调用。没有属性说明的函数，按外部函数对待。

② 函数类型说明 该部分用于定义函数返回值的数据类型。C 程序所使用的基本数据类型有 `int`(整型)、`char`(字符型)、`float`(浮点型)。C 程序里没有类型说明的函数按整型对待。C 程序的函数相当于其他语言的子程序和过程。在 C 语言里，也有人把有返回值的叫函数，无返回值的叫过程。

③ 函数名 函数名是识别函数的名字。有效的名字是以英文字母(`a~z`, `A~Z`)或下划线(`_`)为字头的英文字母、数字和下线符的序列。字母和数字的序列叫标识符，标识符可任意长。

注意：C 语言对字母的大小写视为是不同的字母，即在 C 程序里，函数名相同位置的字母相同而大小写不同的命名，被当成是两个函数。

④ 参数表 参数表在函数名后的圆括号(和)内。这里的参数是形式参数，简称形参，也叫哑元。当然函数被调用时，形参要被实在参数(简称实参，也叫实元)所替换。函数没有形参时，也不能省略掉圆括号(和)；因为圆括号是函数的特征符，只有后接()的标识符才被看做是函数名。前面所遇到的函数 `main()` 就是没有形参的函数。

⑤ 参数类型说明 该部分用于指定参数的数据类型。

有些专家把 C 语言分为传统的和现代的两种。像上述的参数类型说明方式就被说成是传统风格，而在现代风格中，参数类型说明可以放在函数名后面的()中。两种风格对比情况如表 1.3 所示。

表 1.3 传统风格与现代风格对比

传统风格	现代风格
<pre> int max(a,b) int a,b; { ... } </pre>	<pre> int max(int a, int b) { : } </pre>

ANSI 允许在函数定义前先给出函数原型，即预先定义函数的参数类型及函数返回值的类型，这样可以对参数及函数返回值强行进行类型检验。例如，下面的函数 `fun()` 就是被

原型化了的。

```
float fun(int,float);  
main( )  
{  
    :  
}  
float fun(a,b)  
int a;  
float b;  
{  
    :  
}
```

(2) 函数体是进行预定处理的部分。

函数体从左花括号({)开始,直到与其对应的右花括号(})为止。

变量说明通常接在{后面。变量也是用标识符表示的,其基本类型也是 int、float 和 char。

接在变量说明后的是执行语句部分,这部分是实际生成命令码的地方。函数体末是右花括号(}),表示函数体终了。

4. C 程序的显著特点

通过前面的学习,已经了解到 C 程序的一些风格和特点。这里再强调一下,本书认为, C 程序有如下 4 个显著特点。

(1) 程序结构的模块化 C 程序是由文件组成的。文件是 C 程序的编译单位,也就是说 C 程序在编译时,一次只能编译一个文件,而不能是两个或两个以上文件,也不能是文件的一部分。因此,通常,把一个 C 源文件叫一个程序单元。

一个文件是由一个或多个函数组成。函数是 C 程序的基本单位。一个函数,一般用来实现某种功能。

含有主函数 main()的文件,叫做主文件。

一个函数(除最小函数外)是有若干条语句组成的。语句是 C 程序的最小单位。

可见,组成 C 程序的成分,从大到小排列顺序是程序、文件、函数、语句。而 C 程序结构的模块化主要体现在它是由其基本模块——函数组成上。

(2) 书写的小写化 C 语言所定义的关键字、转换控制字符、控制语句,以及所有库函数都是用小写字母命名的,在使用时,也必须用小写。同样在编写程序时,也提倡用小写。这主要是为了照顾人们的习惯。

(3) 语句的表达式化 出现在 C 程序中的任何合理的数学表达式,均可被 C 编译系统编译执行。这说明 C 语言表达能力极强,适用于解决任何领域的问题。也正因为 C 程序可含有大量表达式语句,故有些专家称 C 语言为表达式语言。

(4) 输入输出的库函数化 C 语言没有定义 I/O 语句,其程序是靠有关库函数来实现数据的输入输出功能的。

5. C 程序的执行过程

一个 C 程序的执行要经过编辑、编译、链接、执行 4 个步骤。编译和链接过程如图 1.2 所示。

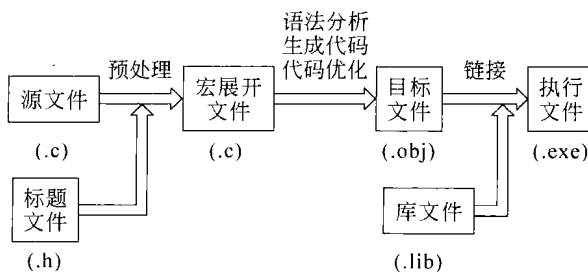


图 1.2 C 程序编译及链接过程

C 编译系统中含有预处理程序,用来处理原文件中带有#的控制行(称为宏指令)。这是编译过程中的第一步。为便于使用,这里,先介绍两条宏指令。

① 宏定义

```
#define 宏名 宏体
```

该宏指令通知预处理程序,把源程序中出现的该宏名都要用所定义的宏体替换掉。例如,若源程序中有如下宏指令:

```
#define INFY -1
```

则该源程序被预处理程序处理后,源程序中的所有宏名 INFY,都将被宏体-1 所置换。

② 宏包含

```
#include "文件名"
```

也可表示为

```
#include <文件名>
```

该宏指令指示预处理程序,用文件名所指定的文件内容替换掉该宏指令,也就是把指定的文件内容嵌入到现行程序中。例如,若源程序中有如下宏指令:

```
#include "stdio.h"
```

则该源程序被预处理程序处理后,文件 stdio.h 的全部内容便嵌入到该源文件中,其内容可以被源文件使用。

1.3 上机操作

Visual C++ 6.0 是美国微软公司(Microsoft)1988 年推出的可视化开发环境,可对 C/C++ 程序进行各种操作,例如,文件的建立、打开、编辑、调试、编译、链接与执行。本书中的练习都是在 Visual C++ 6.0 环境下进行的。本节介绍 Visual C++ 6.0 的运行环境及使用方法。

1. Visual C++ 6.0 的运行环境

Visual C++ 6.0 被启动后,显示屏幕上就会显示出它的运行环境,如图 1.3 所示。

从该图可以看出,Visual C++ 6.0 运行环境是由如下几部分组成。

(1) 标题栏 在显示屏的最上面,从左到右依次是窗口控制图标、窗口标题、“最小化”