

微机C语言基础与应用

王泰峰 编著

中国铁道出版社

550011

微机 C 语言基础与应用

王泰峰 编著



中国铁道出版社

1991年·北京

7552/67

— 内 容 简 介 —

C语言是一种通用的程序设计语言,它以鲜明的特点和独特的设计而受到程序设计者的普遍重视。随着微机被广泛地使用在各个领域,使用C语言开发的微机应用软件倍增。为了适应微机程序设计者的需求,笔者总结了多年在日本编程的实践体会,在对教学讲稿进行几次修改的基础上编写了此书。

本书共分14章。前13章详细分类介绍C语言的基本概念以及C程序的编程方法。第14章的内容是从应用的角度出发,重点介绍进程控制、FAR指针的应用以及C语言与汇编语言接口等一些微机应用软件开发中经常遇到的问题和实际例程。

本书在叙述上由浅入深,循序渐进。书中附有大量的简洁、易懂的实例以及图解说明,使读者很易理解和掌握。除第1、14章之外,每一章后面还提供了一些应用实例及习题,可帮助读者进一步巩固、消化本章所学的内容。书中绝大部分例题无需任何修改均可在不同C系统下编译、运行。本书不但能使读者全面、深入掌握C语言的基本概念,并且能帮助读者提高程序设计技巧,以及在微机上开发应用软件的能力。

本书可供C语言初学者自学,也可作为高等院校的教材及计算机应用方面的技术人员的参考书。



中国铁道出版社出版、发行

(北京市东单三条14号)

责任编辑:殷小燕 封面设计:陈东山

各地新华书店经售

中国铁道出版社印刷厂印

开本:787×1092毫米 1/16 印张:22 字数:546千

1991年7月 第1版 第1次印刷

印数:1—7500册

ISBN7-113-01062-8/TP·107 定价:12.00元

前 言

C 语言最初是为描述 UNIX 操作系统而研制成功的一种新型的程序设计语言,因此它是最适于编写操作系统以及靠近硬件部分的工具软件。又因为 C 语言还具有语言简洁、表达能力强、代码质量高以及可移植性好等特点,所以很快被人们所接受并且得到了广泛的推广和使用。从 C 语言问世到现在,由于其自身的不断发展和完善,C 语言不但可以用来编写系统软件,而且成为一种用于开发应用程序的强有力的工具。目前众多的软件技术人员利用此工具在开发各种应用程序。受此影响,从微机到大型机,各种版本的 C 语言编译系统相继问世。在微机上许多最新的 C 语言版本中,不但保证了与 XENIX 或 UNIX 之间最大程度的兼容性,而且扩充了许多功能强、速度快、能高效地实现程序设计中常见任务的函数,从而节省了程序员自己用来编写这些函数的时间和精力。本书中所介绍和使用的 MS—C 就是其中的一种。随着 C 版本的增多、函数功能的增强,C 语言越来越受到大批计算机用户特别是软件技术人员的欢迎。尤其在日本、美国、加拿大等国家的许多公司,在微机上用 C 语言开发的软件产品越来越多,形成了一种 C 语言“热”。比如,由于 C 语言具有可分别编译及可移植性强的特点,日本的一些程序员,白天在公司的大型机 UNIX 系统下编写 C 程序,晚上仍可在家庭的微机上继续编写、调试。

目前在我国使用微机的用户越来越多,而且众多的软件技术人员是在微机上开发应用程序。为了在微机上进一步推广使用 C 语言,笔者结合自己去日本公司几年研修期间在微机上使用 C 语言开发应用程序的经验和教训,以及在国内多次举办 C 语言学习班时讲授 C 语言的体会,并参阅了许多中、英、日文的有关资料,在讲稿的基础上经过整理写成了这本书。笔者结合书中的例题,把在实际工作中亲身所遇到的经验教训以及使用 C 语言必须要注意的一些重要问题和细节问题提供给读者做为参考。这些问题往往在学习 C 语言时,或是使人感到困惑,或是被人忽视,但在实际应用程序开发工作中遇到后又不知所措。

本书中的所有例题都是在 IBM—PC 及其兼容机上,用 Microsoft C4.0、5.0 编写调试通过的。而且还使用了 Turbo C1.5、2.0 进行编译、调试,除了极个别之处,程序不必做任何改动就可编译、运行(需要改动之处,书中加了说明)。

笔者在编写本书时力求做到以下几点:

1. 由浅入深

本书主要是针对自修 C 语言的初学者而编写的,使读者能够边阅读边上机实习,从而逐步掌握 C 语言。在保持全书的连贯性、论述体系的一致性和科学性的前提下,对论述的内容进行合理分类,力求逐步增加、引深概念。在实例中尽量避免后面章节中所论述的内容提前出现。有些确实避免不了的内容(例如,C 预处理的某些命令等),也加上注释并指出了所在的章节。

2. 通俗易懂

书中尽量避免用大篇的文字来论述一些概念,而是采用大量的图解和例题并且加以注释,同时给出程序的运行结果的方式,使读者从直观上理解和掌握新概念。书中尽量选择了一些能全面代表各种类型而又简单明了、可以单独上机实习的例题程序,使读者通过例题就可以一目了然地理解基本概念和 C 语言程序的设计方法,而不必去花时间搞懂例题中的某些算法和有待解决的具体问题后才能理解所论述的概念。

3. 查阅方便

为了使有一定 C 语言基础的读者,在实际编程中遇到问题时能够方便地查阅所要了解的内容,本书不但按章节详细地列出了目录,而且在书后还提供了英文词、符号的索引。并且把一些在编程工作中经常需要查阅的内容,归纳在附录表中。

4. 便于移植

考虑到读者所持有的计算机系统以及 C 语言版本的不同,书中的例题基本上采用了 ANSI C 中所指定的标准库函数。并且在附录中给出了几种不同 C 版本的库函数对照表。这样,书中的所有例题都可方便的移植到读者使用的系统上运行。

全书共分14章。第1章至第13章可供初学者学习(加有 * 号的章节及例题可跳过不读),第14章中分类介绍了一些使用 C 语言进行高水平编程的方法,可供有一定 C 语言基础的技术人员参考。为了帮助读者学习本书的内容,笔者把书中的习题解答及部分应用例题的源程序,配制在一张360k、5英寸软盘上。有欲购者可与出版社发行部联系。

全书由北京计算机学院副教授王勇领先生审阅。易小琳同志也对本书提出了许多宝贵意见。在编写本书的过程中,还得到了朱东虹、宋燕、付燕杰、夏地、丘正武、杨玉、初征、方蕙等同志的大力帮助,在此一并表示感谢。由于笔者水平有限,书中一定存在不少缺点和错误,恳请读者批评指正。

王 泰 峰

一九九〇年六月 于北京计算机学院

目 录

第1章 C 语言程序概述	1
1.1 C 语言的历史和发展	1
1.2 C 语言的特点	2
1.3 C 语言程序结构	3
1.4 C 语言程序的编译和链接	5
1.4.1 C 程序的开发环境	5
1.4.2 C 程序的编译和链接过程	7
1.4.3 CL 命令的使用方法	8
第2章 基本概念	10
2.1 C 语言的基本元素	10
2.1.1 标识符	10
2.1.2 关键字	11
2.1.3 空白字符	11
2.1.4 注 释	12
2.1.5 分隔符	12
2.2 变量和数据类型	13
2.2.1 变 量	13
2.2.2 数据类型的种类	13
2.2.3 基本数据类型	14
2.3 常 量	17
2.3.1 整数常量	18
2.3.2 浮点数常量	18
2.3.3 字符常量	18
2.3.4 字符串	20
2.4 综合练习	20
2.4.1 按格式输出函数 printf	20
2.4.2 转义序列的应用	22
2.4.3 printf 中控制字符的应用	23
2.4.4 printf 中格式选择项的应用	24
2.4.5 练习题	25

第3章 表达式和运算符	27
3.1 表达式.....	27
3.2 运算符.....	29
3.2.1 算术运算符.....	30
3.2.2 关系运算符.....	32
3.2.3 求反运算符.....	33
3.2.4 移位运算符.....	34
3.2.5 按位运算符.....	37
3.2.6 逻辑运算符.....	38
3.2.7 条件运算符.....	41
3.2.8 赋值运算符.....	42
3.2.9 一元加1、减1运算符	44
3.2.10 sizeof 运算符.....	46
3.2.11 逗号及其它运算符	47
3.3 类型转换.....	48
3.3.1 算术转换.....	48
3.3.2 赋值转换.....	49
3.3.3 强制类型转换.....	50
3.4 综合练习.....	52
3.4.1 应用例题.....	52
3.4.2 练习题.....	53
第4章 结构控制语句	54
4.1 语 句.....	54
4.2 结构控制语句.....	54
4.2.1 while 循环语句	55
4.2.2 do~while 循环语句	57
4.2.3 for 循环语句	58
4.2.4 if~else 条件语句	60
4.2.5 switch 开关语句	64
4.2.6 goto 语句和标号	66
4.3 辅助控制语句.....	67
4.3.1 break 间断语句	67
4.3.2 continue 继续语句	68
4.3.3 return 返回语句	68
4.4 综合练习.....	68
4.4.1 应用例题.....	68
4.4.2 练习题.....	71

第5章 数 组	73
5.1 数组的基本概念	73
5.2 一维数组	74
5.3 字符串与字符数组	76
5.4 多维数组	78
5.5 综合练习	79
5.5.1 应用例题	79
5.5.2 练习题	81
第6章 函 数	83
6.1 函数的结构	83
6.2 函数的形参与实参	86
6.3 函数的说明与定义	88
6.4 void 类型	90
6.5 函数的递归调用	90
6.6 综合练习	92
6.6.1 应用例题	92
6.6.2 练习题	95
第7章 存储类别	97
7.1 变量的存在性和可见性	97
7.2 变量的存储类别	97
7.2.1 自动变量	98
7.2.2 寄存器变量	99
7.2.3 外部变量	100
7.2.4 静态变量	103
7.3 变量的初始化(小结)	104
7.4 函数的存储类别	105
7.5 综合练习	107
7.5.1 应用例题	107
7.5.2 练习题	109
第8章 指 针	110
8.1 指针与地址的概念	110
8.2 指针的种类	112
8.3 指针与数组	113
8.3.1 指针与一维数组	113
8.3.2 指针与多维数组	115
8.4 字符串指针	116

8.5	指针运算	119
8.6	指针类型的转换	121
8.7	指针数组	121
8.8	多级指针	126
8.9	综合练习	128
8.9.1	应用例题	128
8.9.2	练习题	131
第9章	函数与指针	133
9.1	函数的指针参数	133
9.2	命令行参数	139
9.3	返回指针的函数	140
9.4	指向函数的指针	141
9.5	函数中的函数指针参数	142
9.6	函数指针数组	143
9.7	复杂说明的理解方法	144
9.8	综合练习	146
9.8.1	应用例题	146
9.8.2	练习题	151
第10章	结 构	153
10.1	结构的基本概念	153
10.2	结构数组	159
10.3	结构指针	160
10.3.1	指向结构的指针	160
10.3.2	结构指针与结构数组	162
10.3.3	结构指针运算	163
10.4	结构与函数	165
10.4.1	结构参数的传递	165
10.4.2	返回结构类型的函数	168
10.5	结构的嵌套	171
10.6	递归结构	173
10.7	包含位域的结构	177
10.8	类型定义 typedef	181
10.9	综合练习	181
10.9.1	结构数组的传递	181
10.9.2	队列和栈	183
10.9.3	双向链表	186
10.9.4	二叉树	188
10.9.5	练习题	194

第11章 联合、枚举类型	195
11.1 联 合	195
11.2 结构中嵌套联合	197
11.3 联合中嵌套结构	199
11.4 枚 举	202
11.5 综合练习	206
11.5.1 应用例题	206
11.5.2 练习题	210
第12章 C 预处理程序	211
12.1 宏替换命令	211
12.2 包含文件命令	213
12.3 条件编译命令	214
12.3.1 #if~#elif~#else~#endif 命令	215
12.3.2 #ifdef(ifndef)~#else~#endif 命令	217
12.4 行控制命令	218
12.5 综合练习	219
12.5.1 应用例题	219
12.5.2 练习题	220
第13章 输入和输出	221
13.1 标准输入/输出	221
13.1.1 标准文件与 FILE 结构	221
13.1.2 字符的输入/输出	223
13.1.3 字符串的输入/输出	224
13.1.4 按格式输入/输出	225
13.1.5 内存数据的格式转换	229
13.1.6 I/O 的重新定向	231
13.2 文件的存取	232
13.2.1 文件的打开与关闭	232
13.2.2 错误处理与出口	235
13.2.3 文件的读/写操作	236
13.2.4 文件的随机存取	239
13.3 低级 I/O	242
13.3.1 文件的读/写操作	243
13.3.2 文件的打开、关闭和创建	244
13.3.3 文件的随机存取	247
13.4 有关文件操作的其它函数	247
13.5 综合练习	248

13.5.1 应用例题.....	248
13.5.2 练习题.....	254
第14章 C语言的高水平编程	256
14.1 进程控制.....	256
14.1.1 system 函数	256
14.1.2 spawn 函数	257
14.1.3 子进程的环境设置.....	260
14.2 far 指针的应用	263
14.2.1 near 指针和 far 指针	263
14.2.2 MS-C 中段和组的分配	264
14.2.3 跨段的数据访问	265
14.3 C语言与汇编语言的接口	267
14.3.1 基本概念和规则.....	267
14.3.2 传递整型参数、返回整型值	270
14.3.3 传递字符串.....	271
14.3.4 传递指针.....	272
14.3.5 传递长整型参数、返回长整型值	274
14.3.6 从C访问汇编中定义的数据	275
14.3.7 在C程序中嵌入汇编子程序	277
14.3.8 far 函数调用	280
14.4 C对其它高级语言的调用	281
14.4.1 C与其它高级语言的接口	281
14.4.2 C与其它高级语言之间的数据共享	281
14.4.3 应用例题.....	282
14.5 图形处理.....	284
附录A MS-DOS 标准 ESC 序列	291
附录B C语言库函数	296
附录C MS-C 编译和链接命令.....	317
附录D MS-C 编译、链接错误信息一览表	318
表索引.....	337
英文、符号索引	338
参考文献.....	341

第1章 C 语言程序概述

C 是一种通用性很强的结构程序设计语言，它具有丰富的运算符、经济实用的表达式、先进的控制结构和数据结构，而且表达能力强、语言简单灵活，还可以生成有效、紧凑和可移植的代码。

1.1 C 语言的历史和发展

C 语言是由美国 AT&T(American Telephone & Telegram)贝尔(Bell)实验室的 D. M. Ritchie 在 UNIX 系统上研制成功的。C 语言于 1972 年正式投入使用。1973 年 K. Thompson 和 D. M. Ritchie 为美国 DEC 公司的 PDP-11 计算机用 C 重写了 UNIX 操作系统。因此，C 语言既与 UNIX 系统有十分密切的关系，但又独立于 UNIX 系统。

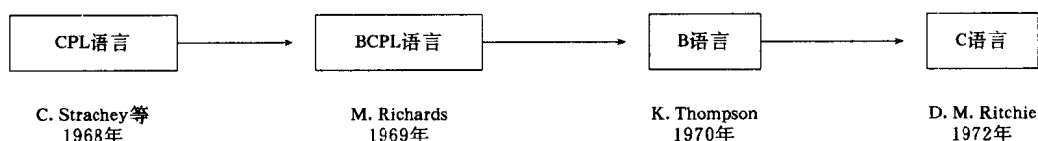


图 1.1 C 语言的演变历史

图 1.1 中图示了 C 语言的演变历史。CPL 语言是以语法规规范化、构造模块化为思想基础，以能够方便高效地描述程序为目的而被开发出来的。为了更有效地描述系统程序，1969 年 M. Richards 在 CPL 的基础上研制了 BCPL 语言。B 语言的主要思想又是起源于 BCPL，于 1970 年，K. Thompson 在 PDP-11/20 上实现了 B 语言并用 B 写了 UNIX 操作系统。由于 BCPL 和 B 两种语言通用范围小等原因，没有被普及使用，但是 BCPL 和 B 的设计思想对 C 语言的研制起了很大的作用。用 C 语言重写后的 UNIX 可读性好、可移植性高，加之 UNIX 本身的优点，使 UNIX 成为国际上使用最为广泛的操作系统。UNIX 的广泛流传又进一步扩大了 C 语言的影响。

在许多厂家的计算机上，各种 C 编译系统也相继问世。例如，IBM VS/370、Honeywell 6000、Interdata 8/32、IBM-PC 及其兼容机、以及日本的 PC-9801 系列机等都有了各种不同版本的 C 编译系统，从而又推进了 C 语言的普及和应用。

目前在世界上出售的 C 版本很多，如，Whitesmiths' C、ai-C86、LSI C、Supersoft C、OPTIMIZING C86、Ci-C86、AZTEC C86、DeSmet C、BDS-C、NICE C、Lattice C、MS-C、Turbo C 以及 C++ 等。美国和日本广泛使用的是 Lattice C、MS-C 和 Turbo C。

为了使 C 语言标准化，美国 ANSI(the American National Standards Institute)从 1983 年夏开始着手制定了 C 语言的标准方案。在此以后的许多 C 的新版本都参照了此标准，这样就给 C 语言程序的移植创造了更有利的环境。

1.2 C 语言的特点

操作系统是系统程序的典型代表，是与计算机硬件有着紧密联系的程序。由于对它的性能要求很高，所以不仅过去，而且现在普遍是用汇编语言来描述操作系统的。C 语言就是为取代汇编语言来描述操作系统而被开发研制的，它不但向用户提供了接近硬件及在比较低级水平上使用计算机的能力，而且有效地解决了汇编语言程序在可读性和可移植性上存在的问题。因此有人称 C 语言是一种“高级汇编语言”。

为了说明 C 语言的特点，在图 1.2 中，我们把 C 语言与其它几种高级语言进行了比较。

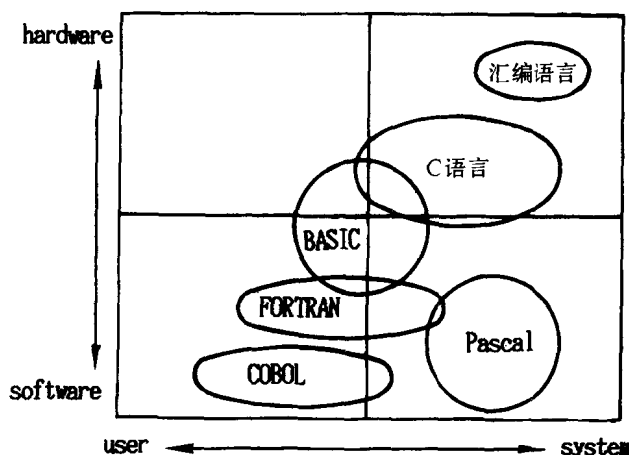


图 1.2 C 语言与其它程序语言的比较

FORTRAN、COBOL 及 Pascal 等高级语言都不能直接对机器硬件部分进行操作（当然这也是它们的优点），因此在一些应用中，用 C 可以较容易实现的一些功能而用这些语言则很困难。

C 语言的特点大致分为以下几个方面：

1. C 是一种编译型语言

经过编辑后得到的 C 源程序，必须经过 C 语言编译程序转换成与它等价的、计算机可直接执行的机器语言程序，即目标程序。一个较大的 C 程序，可以分开存放在若干个源文件中，分别编译成功后再链接在一起，形成一个可执行程序。

2. 程序书写简洁、清晰

一个 C 语言程序由若干个函数定义的集构成。C 中的函数（独立的子程序）提供了编制结构化程序的手段，使得程序结构清晰并易于阅读和维护。C 语言中提供了丰富的运算符，使用灵活、易于掌握，可以使程序书写简洁而且执行效率高。“Small is beautiful”这句话用于评价 C 语言恰如其分；一个短小的 C 程序可处理复杂的问题。

3. 语言表达能力强、通用性好

C 语言提供了指针、地址以及位操作等功能，可以直接存取内存中的数据以及完成通常由硬件实现的算术、逻辑运算。这是 C 语言有代表性的功能。在多数情况下，一个优秀的 C 语言程序在速度上可与汇编程序相比，它足以有效地取代汇编语言来编写各种软件。例如，C 语言编译程序本身就是用 C 语言编写的。C 是一种适用于从 8 位微型机到大型计算机的通用性

语言，用它不但可以编写操作系统、编译程序等系统软件，而且可以编写 CAD、数据处理、过程控制、信息通讯以及人工智能、解计算机病毒等各种应用软件。例如，许多传真机、复印机上的控制软件就是用 C 语言编写的。

4. 先进的控制结构和数据结构

C 语言提供了一整套循环、条件判断和转移语句，实现了对程序逻辑流的有效控制；有利于结构化程序设计。同时，还提供了丰富的数据构造类型（如：数组、结构和联合等），用以实现对复杂的数据类型的描述。

5. 代码质量高，可移植性好

C 语言中大多数运算符是与一般机器指令相一致的，可直接翻译成机器代码。另外，一些运算符明确地指定了不同的操作，可产生最短的机器代码。因此在代码质量上 C 可与汇编语言相媲美。C 语言程序可以从某一环境不加或稍加改动就可搬到另一个完全不同的环境上运行，这是最为人们所称赞的一个特点。C 还提供了预处理功能，在编译之前可根据源文件中的预处理命令要求，对源文件进行再安排。预处理命令最典型的作用是使源文件与执行环境有关的部分得到修改和选择，从而使一个 C 语言程序，适应不同的执行环境。

6. 具有多种标准存储模式

不同的 C 版本，存储模式的种类多少有些差别，但一般来说都具备下述三种模式：

- 小模式 (Small model)

程序代码段小于 64K 且程序数据段小于 64K 时使用。对于大多数应用程序而言是足够用的了。

- 中模式 (Middle model)

程序代码段大于 64K，而程序数据段小于 64K 时使用。

- 大模式 (Large model)

可以建立多代码段和多数据段。

程序员可以根据具体要求选择模式来编译 C 程序，以建立最佳的代码存储和数据存储的组合，从而提高效率。

7. 不具备输入/输出设施

C 语言中没有提供输入/输出（如：READ、WRITE）语句，也没有现成的访问文件的指令。所有 I/O 功能都是通过调用库函数来完成的。C 提供了大量而有效的库函数，而且这些库函数可根据需要方便地扩充。

1.3 C 语言程序结构

C 语言程序一般是由一个或多个函数组成，它们在程序中出现的顺序可以任意，而且可以分别驻留在一个或几个源文件中。组成一个程序的若干函数中必须有而且只能有一个名为 main 的主函数，它是 C 程序运行开始时被调用的第一个函数。通过 main 函数，直接或间接调用其它函数来控制整个程序的流程。

下面结合图 1.3，简单介绍一下 C 程序的基本结构，程序中的各个组成部分，我们将在后面各章节中还要详细论述。

1. 函数

一个函数是由函数头和函数体构成（详见第 6 章函数）。由函数名及一对圆括号“(”和“)”

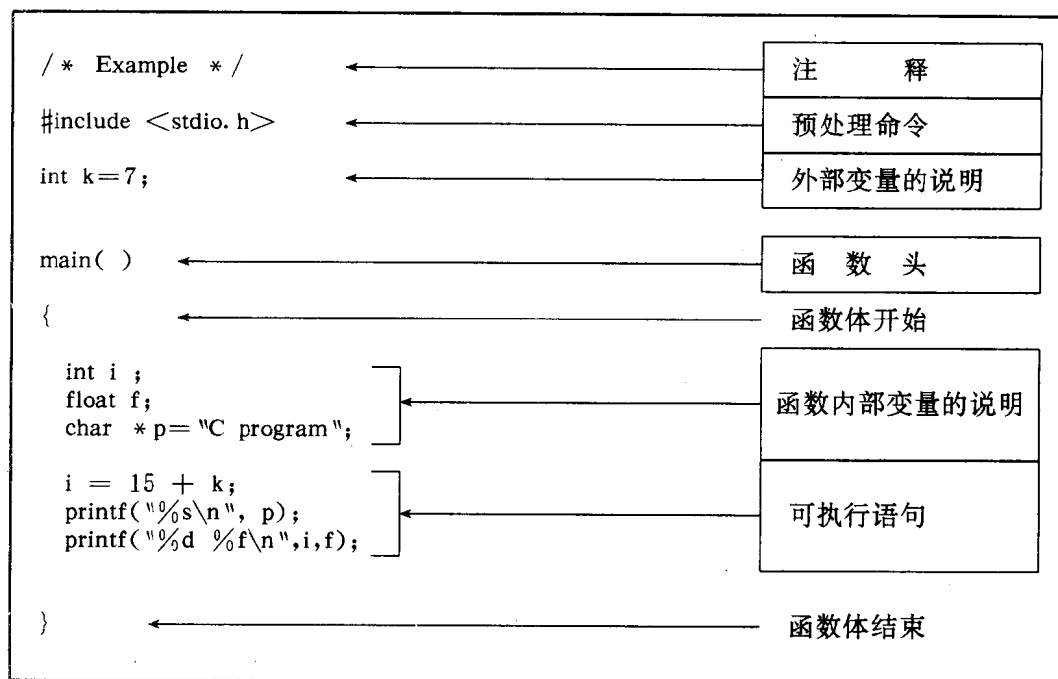


图 1.3 C 程序的基本结构

组成函数头，这是函数的标志(如图 1.3 中的 `main( )`)。

用一对花括号“{”和“}”括起来的部分称为“函数体”(程序体)。函数体中可以有变量说明以及任何有意义的可执行语句。变量说明必须出现在所有可执行语句之前；函数体也可以为空，但花括号不可缺省。函数体中仍可出现由成对的花括号将若干语句括起来的复合语句(称之为分程序)。

2. 语句结束符

在 C 中使用分号“;”作为一个可执行语句、说明语句的结束符。

3. 预处理命令

在 C 语言程序中若调用了 C 的某个库函数，那么必须用预处理命令：`#include` 将包含该函数说明的头文件(head file)嵌入 C 源文件；否则编译就会出错。此命令一般写在源文件的开头(详见第 12 章)。例如，在图 1.3 的程序中，由于调用了库函数 `printf`，所以必须在源文件的头部写有 `#include <stdio.h>` 命令，这是因为 `printf` 的说明包含在 `stdio.h` 文件中。

4. 注 释

C 语言中以“/*”开头，到“*/”结束表示是一个注释(详见 2.1.4 节)。

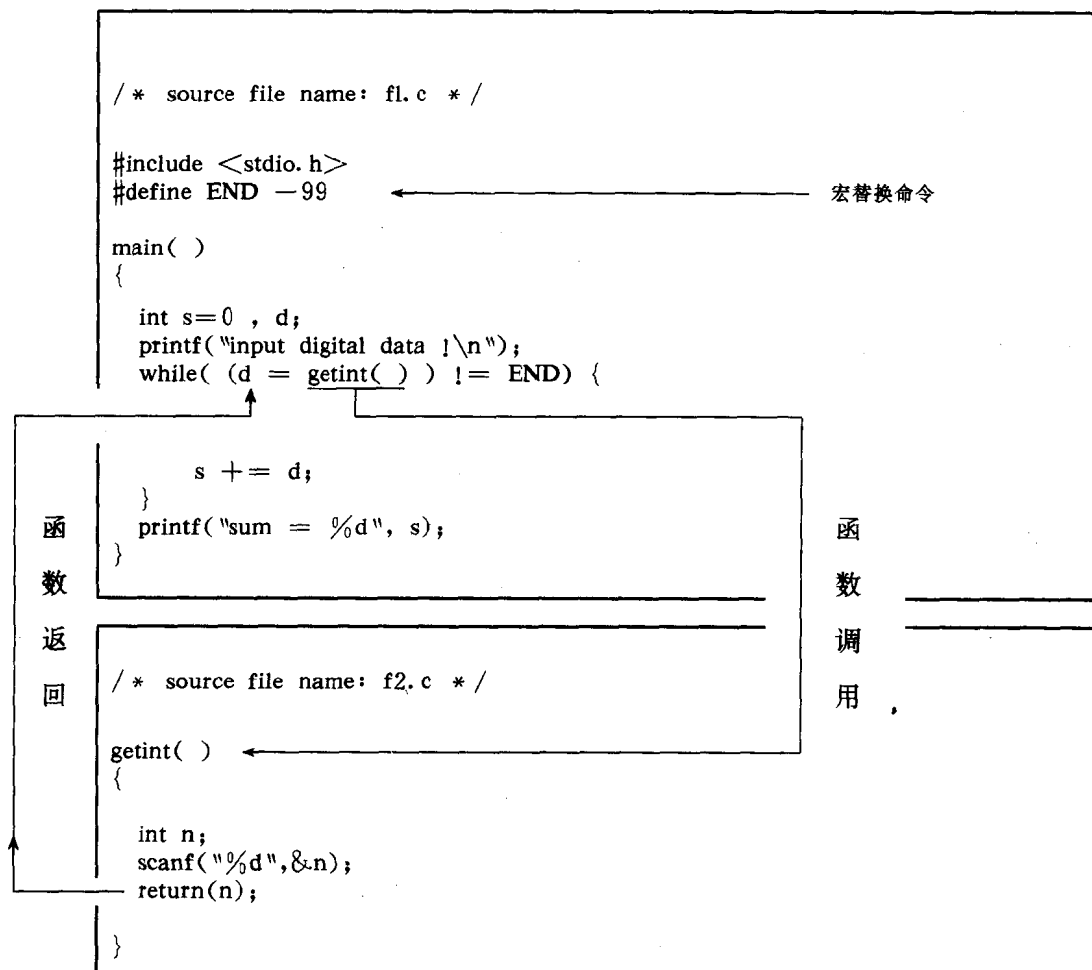
5. 源文件

一个 C 语言源程序可以被分割驻留在一个或多个源文件中。C 源文件是一个正文文件，在该文件中可以包括任意个完整的语句、命令、注释、变量及函数的说明或定义，它们之间可以插入任意个空格或空行，但它们本身不能被分割在两个源文件中。源文件的最后一个字符必须是换行符。C 语言程序源文件名都以“.c”作为扩展名。

例 1—1 中包括了两个 C 源文件：`f1.c` 和 `f2.c`，在 `f1.c` 文件中调用了一个函数 `getint( )`，而此函数驻留在 `f2.c` 文件中。

源文件中也可以不包括任何可执行语句。例如，可以把程序中使用的宏命令、外部变量的说明等单独写在一个源文件中集中管理，这样做便于寻找和修改。当源程序被编译时，可对

[例 1-1]



每个源文件进行单独编译,然后再将有关的文件(目标文件)链接在一起,形成可执行文件。

1.4 C 语言程序的编译和链接

为了使读者边学习边上机实习,本节中主要介绍在微机上开发 C 语言应用程序的简单过程以及必须具备的最基本的软、硬件环境。

1.4.1 C 程序的开发环境

由于 C 是一种通用性语言,而且具有可移植性的特点,因此从最小的微型机到类似 Cray-2 的大型机,C 语言在大多数机器上均有实现。本书中的例题是在下述环境下调试、运行的:

1. 硬件

- IBM-PC 兼容机(带有硬盘)
- 打印机

2. 软件

- MS-DOS(PC-DOS) Ver 3.0
- Microsoft C 5.0 最优化编译语言系统(以下简称 MS-C 5.0)
- Microsoft 5.0 宏汇编

[注] 读者也可以使用 Turbo C、Lattice C 及 MS-C 等其它 C 版本。

MS-C 是美国 Microsoft 公司研制的、适用于 MS-DOS(PC-DOS)操作系统的优化 C 语言编译系统。它是对 C 语言作者 Brian W. Kernighan 和 Dennis M. Ritchie 在《The C Programming Language》一书中定义的 C 语言的一个完全实现,被指定为美国 89 年国家标准。Microsoft 公司积极参与了 ANSI 对于 C 语言美国国家标准的研制。MS-C 5.0 充分照顾了这一最新的 C 标准,并且努力与之保持一致。在设计 MS-C 库函数时,保证了它们在 MS-DOS 与 UNIX (XENIX) 系统之间最大程度的兼容性。因此,本书以 MS-C 5.0 为主,介绍微机 C 语言的基本概念及其应用。考虑到读者所具备的软、硬件环境的不同,除第 14 章之外,绝大部分例题都是按 ANSI 标准 C 的要求编写的,读者可在不同的环境下编译、运行(这些例题均用 MS-C 4.0、5.0 及 TURBO C 1.5、2.0 调试通过)。在附录 B 中还给出了几种不同 C 版本的库函数对照表,供读者参考。

在开始学习 C 语言之前,首先应建立一个实习 C 语言的环境。下面我们以 MS-C 和 TURBO C 为例,简单介绍如何在磁盘上设置 C 程序的编译及链接环境的方法。

1. 如果读者所使用的微机带有硬盘,可按如下步骤将 MS-C 5.0 编译软件装入硬盘。

(1) 将带有 SETUP.EXE 程序的第一张软盘插入软盘驱动器 A 中,并键入如下命令:

```
A>SETUP C:\MSC S
```

只装入小模式

在硬盘的根目录下建立一个子目录: MSC

然后按照 SETUP 程序的提示信息,顺序将各软盘中的 MS-C 编译软件装入硬盘。

(2) 修改 DOS 系统中的 CONFIG.SYS 和 AUTOEXEC.BAT 文件的内容。

• 在 CONFIG.SYS 文件中加入如下内容:

```
FILES=20
BUFFERS=10
DEVICE=ANSI.SYS
```

• 在 AUTOEXEC.BAT 文件中加入如下内容:

```
PATH=c:\msc\bin
set INCLUDE=c:\msc\include
set LIB=c:\msc\lib
set TMP=c:\msc\tmp
```

这两个文件修改后,需重新启动 DOS 系统。进入 MSC 子目录运行 QC.EXE,即可编辑、编译及运行 C 程序。

2. 如果读者所使用的微机仅带有两个软盘驱动器,可使用 MS-C 的 Quick C 或 TURBO C。下面以 TURBO C 2.0 为例,简单介绍如何将编译软件装入软盘的方法。

首先准备三张已格式化的软盘,然后在驱动器 A 下插入带有 INSTALL.EXE 程序的软盘,并键入如下命令:

```
A>INSTALL
```