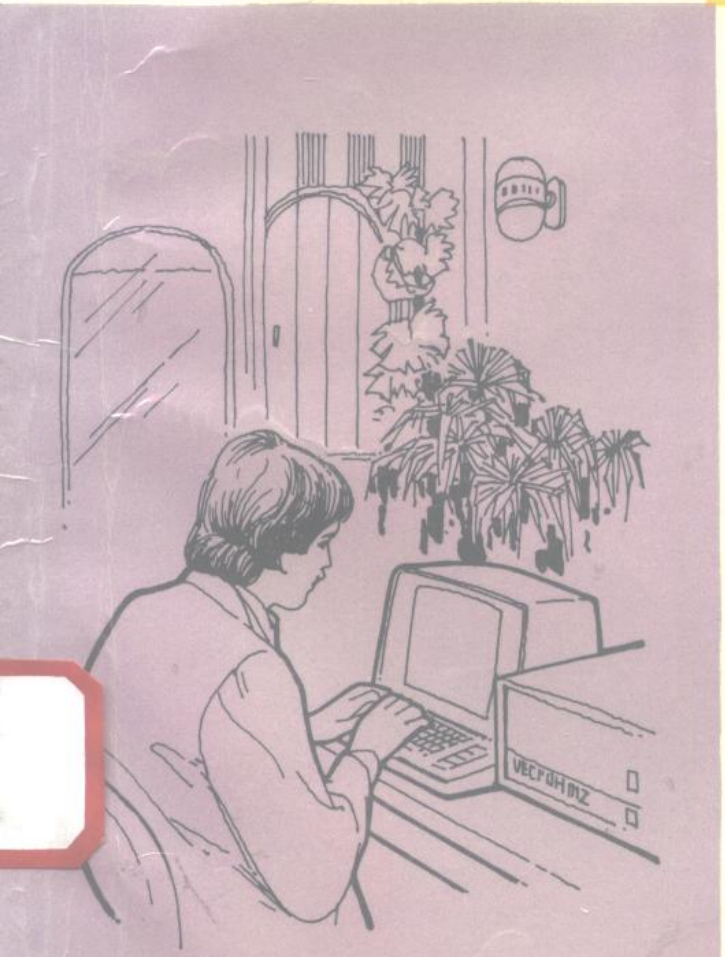
计算机应用入门丛书

C语言实践入门与上机操作
(MSC 5.0篇)

陈　进等编著



C YUYAN SHIJIAN RUMEN YU SHANGJICAOZUO (MSC 5.0 PIAN)

上海交通大学出版社

计算机应用入门丛书

C语言实践入门与上机操作
(MSC 5.0篇)

陈　进　汪东海
王小铁　杨　扬
编　著

上海交通大学出版社

内 容 提 要

本书以符合 ANSI C 标准的 Microsoft C 5.0/Quick C 1.0 为背景，从 C 语言的基础知识开始，介绍了微机版本 C 语言的系统安装、程序设计及操作入门、以及程序调试和优化编译的基本方法和技巧。全书力求由浅入深，为具有一定 C 语言基本知识，但又缺乏上机经验的读者提供一个快速的上机实践参考手册。

为了帮助读者尽快掌握 C 语言及上机实践的基本方法，本书还专门准备了一套指导上机实践的练习题集存放于软盘。需要该软盘的读者可与本社计算机编辑室联系。

读者对象：大专院校师生、工程技术人员、计算机应用工作者。

(沪)新登字 205 号

C 语言实践入门与上机操作

出版：上海交通大学出版社

（上海市华山路 1954 号　邮政编码：200030）

发行：新华书店上海发行所　　印刷：立信常熟印刷联营厂

开本：850×1168(毫米)1/32　　印张：3.875　字数：102,000

版次：1995 年 3 月　第 1 版　　印次：1995 年 4 月　第 1 次

印数：1—6000　　科目：342—269

ISBN 7—313—01410—4/TP・261　　定　价：5.00 元

前　言

C 语言作为一种通用的程序设计语言，早已不再是只由专业的程序员掌握的程序设计工具了。早期，C 语言主要被用来设计 UNIX 操作系统，它是和 UNIX 操作系统相辅相成地发展起来的，是一种极好的系统语言。近年来随着微型计算机的深入普及和微机版本 C 语言的完善，C 语言也成了一种很好的应用语言，受到了越来越多的非专业程序设计人员的喜爱。之所以如此，主要是因为 C 语言在保证高级程序设计语言的结构、流程控制和编程环境的条件下，还提供了类似于宏汇编语言那样的系统资源操作能力和程序执行效率。

然而，C 语言目前在我国的普及程序却相对较低，应用也不够深入，过去在大学的非计算机专业本科课程中也很少开设 C 语言课程。因而在一定程序上阻碍了我国软件和计算机应用水平的发展和提高。同时，由于 C 语言实在是内容太丰富、功能太强大、使用太灵活了，也就是说，对于稍有程序设计经验的人来说，学会简单的编程和应用并非难事，但要真正掌握 C 语言的精髓，充分利用 C 所独具的特点，发挥其潜力，则比较困难，可谓“入门容易得道难”。解决这个问题的最有效的办法就应该是上机实践！因为学习计算机程序设计语言除了基本知识之外，在很大程序上还取决于上机的经验。

为此，我们编写了这本供上机实践使用的类似于参考手册性质的书，其主要目的是为那些初次接触 C 语言或稍有一些 C 语言知识的读者提供一个上机入门的辅助指导。

本书以符合 ANSI C 标准的 Microsoft C 5.0/Quick C 1.0 为背景，采用循序渐近的方法，逐步将读者带入 C 语言实践的应用环

境，并引导读者尽快全面地掌握利用专门的语言环境进行程序设计和调试的基本方法和技巧。全书共分为五章，其中：

第一章概要地介绍了C语言的部分基础知识。包括C语言的发展史及其特点，C语言的关键字、数据类型与存储种类、运算符、流程控制语句、函数以及指针等。

第二章分析了MSC 5.0/QC 1.0的特点以及系统安装、运行环境建立等上机准备工作。

第三章则重点描述了在QC集成程序设计环境中进行编程、程序调试和维护的具体方法及操作步骤。

第四章说明了利用QCL和CL进行程序优化编译的方法，以及利用MAKE实用程序进行程序自动维护、更新的方法，为设计大型的应用程序提供了一个方便的方法。

第五章给出了MSC 5.0/QC 1.0的各类常用库函数的功能、调用参数和使用方法，为读者提供了一个快速的库函数查阅手册。

此外，在附录中列出了ASCII码和扩展ASCII码表，还给出了二进制、八进制、十进制和十六进制等C语言中经常使用的数字进制之间的转换关系表。

由于作者水平有限，难免会有疏漏与不妥之处，恳请广大读者及有关专家提出宝贵意见。

编者

1994.10.2

目　录

第一章　C语言基础知识

§1—1　概述

C语言是一种通用的程序设计语言，由Dennis M、Ritchie发明并形成于1972年。C语言最初是在配备有UNIX操作系统的PDP—11小型计算机上实现的，之后UNIX操作系统本身以及运行其上的绝大部分软件都改用或采用C语言编写。C语言的起源过程可概括成图1—1。

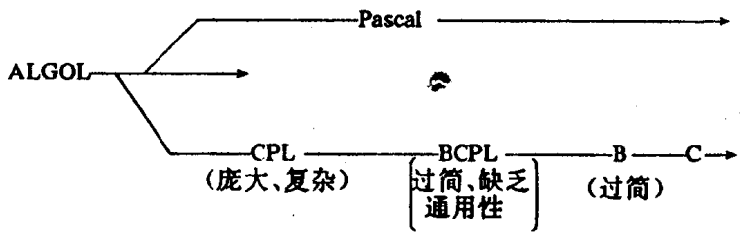


图1—1　C语言的产生过程

尽管C语言是与UNIX操作系统相辅相成地发展起来的，但它早已是一种不单能在UNIX环境下工作的语言了。尤其是在微型计算机广为普及的今天，微机版本的C语言目前已具备了极其强大的功能，其中最具代表的是Microsoft C、Turbo C、Lattice C等等。

C语言是高级语言中一种比较"低级"的语言，也有的专家直接称之为"中级"语言。之所以如此，是因为C语言将高级语言的成分、形式同汇编语言的功能比较完美地结合了起来。在众多的编程语言中，C语言可以说是独一无二的，它一方面能像FORTRAN，Pascal这类高级语言那样为使用者提供方便和自然，另一方面又可以像汇编语言一样允许用户自由地控制硬件及其他外围辅助设

备，并且更加直观、便利。

C语言具有非常多的优点，概括起来有如下几点：

(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。其关键字在标准情况下只有约32个。

(2)C语言非常容易移植。这一方面是由于C语言本身比较小，同时也由于C语言本身并没有输入输出功能，而输入输出都是通过库函数实现的。

(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，也允许使用指针和地址，具备许多近似于汇编语言的功能(如支持位操作等)，在很大程度上可以替代汇编语言进行程序设计。

(4)C语言的基本构件就是函数，函数的集合构成了模块化的结构，换句话说，C语言是一种结构化和模块化的语言。

(5)C语言具有最新的流程控制结构，从而进一步保证了程序的结构化、清晰化，并便于程序的更新、维护。

(6)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，并且各种类型数据之间的转换也十分自由。

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

(8)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，由于C语言是属于“程序员的语言”，在除了具有如上所述的模块化结构，独立函数集合，简洁、紧凑等特点之外，很少给出使用限制和强求，例如没有数组边界自动检查功能等。这样，无疑可以获得高效的执行代码，但另一方面，尤其是对许多“非专业程序员”来说，则往往容易引起一些不必要的错误，稍不注意有时还会导致意想不到的结果。因此，在使用C语言进行程序设计时，应充分注意C语言的特点。

§1—2 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。在程序设计中，C语言的关键字不允许被重新定义或作为其他用途使用。表1—1列出了ANSI C标准所规定的32个关键字。注意，所有关键字都是小写的。

表1—1 C语言的关键字

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

此外，在C语言中还规定了一些“特定字”，如表1—2所示，尽管它们并非关键字，但鉴于它们已有特定的含义，建议用户在定义自己的标识符时最好不要使用它们。

表1—2 C语言的特定字

define	undef	include	ifdef	ifndef
endif	line	error	elseif	pragma

§1—3 数据类型与存储种类

1. 数据类型

C语言提供了五种基本数据类型，即：字符型、整型、浮点型、双精度型和无值型。并且，在某些限定词的修饰下，还可以得到多

种扩充数据类型。一般来说，各种数据类型的长度和值域范围会随处理器的类型和C语言编译器的不同而变化。表1—3中给出了在IBM PC微机上(基于8086,80286,80386等的机型)C语言常用数据类型的字长和值域范围。

表1—3 C语言(微机版)中常用数据类型(ANSI标准)

序号	数据类型	字长(bytes)	值域范围
1	char	1	—128～127
2	int	2	—32,768～32,767
3	float	4	大约6位十进制有效数字
4	double	8	大约14位十进制有效数字
5	unsigned char	1	0～255
6	signed char	1	—128～127
7	unsigned int(unsigned)	2	0～65,535
8	signed int(int)	2	—32,768～32,767
9	short int(short)	2	—32,768～32,767
10	unsigned short int (unsigned short)	2	0～65,535
11	signed short int(short)	2	—32,768～32,767
12	long int (long)	4	—2,147,483,648～2,147,483,647
13	unsigned long int (unsigned long)	4	0～4,294,967,296
14	signed long int(long)	4	—2,147,483,648～2,147,483,647
15	long double(double)	8	大约14位十进制有效数字

void类型通常情况下有两种用法，第一种是用来定义或说明无返回值的函数，第二种用法是用来设置将可能是任意类型的指针。

此外，C语言还提供聚集数据类型，包括结构体(struct)，联合体(union)，位域(bit field)、枚举(enum)以及用户定义的数组类型等等。

2. 存储种类

在C语言中，任何一个变量和函数都具有两个属性，即类型和存储种类。变量的存储种类决定了变量在存储单元中的存储位置，自然这就决定了变量的有效范围或寿命。变量的存储种类可概括为：

变量
- 内部变量
  - 自动变量(auto)
  - 静态变量(static)
  - 寄存器变量(register)
- 外部变量(extern)
  - 全局变量
  - 静态变量

其中，自动(auto)存储种类是一种最为常用的存储种类。在函数内部，当存储种类缺省时，变量均被解释成自动存储种类的变量，这种变量一般是被分配在堆栈之中；寄存器(register)存储种类可以认为是自动存储种类的一种特殊情况，其变量在可能的条件下被分配在高速的寄存器中，当然定义寄存器变量只不过是给编译器的一个建议，是否能够真正取得寄存器变量则要由编译器根据实际情况来确定；静态(static)存储种类通常将变量存放于计算机内存的某一固定位置，静态变量有两个典型用途，其一是为局部变量重新进入定义它的函数或复合语句提供初值，其次是外部静态变量可以提供模块化的"私有机制"，即外部静态变量只能被定义它的模块内的各个函数所共享；最后是外部(extern)存储种类，它主要用于说明在函数外部或别的模块文件中定义过的外部变量，外部变量也被安排在内存中某一固定位置，它可用作为不同函数间的公用参数。

§1—4 运算符

在C语言中，大约有12种类型的运算符，这些运算符从不同的角度使C语言对数据的表达、处理与操作能力得到了极大的丰

富。

1. 赋值运算符

'='被称为赋值运算符，由该运算符可组成普通的赋值语句。

〈变量〉=〈表达式〉;

其意义是先计算出"="右边表达式的值，然后赋值给("搬迁至")左边的变量，在C语言中允许多重赋值，例如：

x=5; y=5; z=5;

可简写成：

x=y=z=5;

2. 算术运算符

算术运算符主要用于进行加、减、乘、除、取模以及取负号等运算。其形式为：

〈表达式1〉算术运算符〈表达式2〉;

表1-4列出了各算术运算符的功能和优先级。

表1-4 算术运算符的功能和优先级

运算符	功 能	优先级
-	负号	↑ 高
*、/、%	乘法、除法、取模	
+、-	加法、减法	↓ 低

3. 关系运算符

关系运算符主要用于在条件判断时，对两个表达式之间的相对关系进行比较。一般情况下多与if，do-while，for等控制语句配合使用。关系运算的形式为：

〈表达式1〉关系运算符〈表达式2〉;

表1-5列出了关系运算符的功能和优先级。关系运算的结果

只有 0 和 1 两种值。当所指定的条件成立时其结果为 1(真),反之,条件不成立则结果就为 0(伪)。

表 1—5　关系运算符的功能和优先级

运算符	功　能	优先级
>、>=、<、<=	比较是否:大于、大于等于、小于、小于等于	↑ 高
==、! =	比较是否:等于、不等于	低

4. 逻辑运算符

逻辑运算符包括对条件式取逻辑非、求两个条件式的逻辑与及逻辑或等。其运算形式为:

(1)逻辑非　　逻辑非运算符〈条件式〉;

(2)逻辑与/或　　〈条件式 1〉逻辑与/或运算符〈条件式 2〉;

表 1—6 列出了逻辑运算符的功能和优先级。其中逻辑非运算符对〈条件式〉的逻辑值取非(反)。逻辑与运算符规定〈条件式 1〉和〈条件式 2〉均为真时结果才为真,它首先对〈条件式 1〉进行判断,当结果为真时就继续对〈条件式 2〉进行判断,反之,若〈条件式 1〉结果为伪时,就不再判断〈条件式 2〉的值了。逻辑或运算符只要两个条件式中的任何一个为真时其结果便为真,当两个条件均不成立时,其结果才为伪。

表 1—6　逻辑运算符的功能和优先级

运算符	功　能	优先级
!	逻辑非	↑ 高
&&	逻辑与	
||	逻辑或	低

5. 增量与减量运算符

增量与减量运算符是 C 语言中十分具有特色和代表性的运

算符，其运算形式为：

(1)前置　　增/减量运算符〈整型变量〉；

(2)后置　　〈整型变量〉增/减量运算符；

增/减量运算符若为前置，则表示首先对整型变量的值进行变更(增加 1 或减少 1)，然后再使用该变量，反之，若为后置，则要在使用过该整型变量之后再对它的值进行变更。表 1－7 列出了增/减运算符的功能。

表 1－7　增量与减量运算符的功能

运算符	功　能
＋＋	将整型变量的值增加 1
－－	将整型变量的值减少 1

6. 位运算符

位运算符也是 C 语言中最具特色的运算符之一，它们具备了与汇编语言相同的功能，包括按位进行与、或、异或、取及以及左、右移位等 6 种运算符，它们只对整型(包括字符型)变量有效。其形式如下：

〈变量 1〉位运算符〈变量 2〉；

表 1－8 列出了位运算符的功能和优先级，同时，在表 1－9 中给出了位运算的逻辑真值表。

表 1－8　位运算的功能和优先级

运算符	功　能	优先级
～	按位取及(求补)	↑ 高
〈,〉	左移位，右移位	
&	按位与(AND)	
∧	按位异或(XOR)	↓ 低
|	按位或(OR)	

表 1－9　按位与、或、异或的逻辑真值表

a	b	与(a&b)	或(a|b)	异或 (a∧b)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

7. 复合赋值运算符

复合赋值运算符由基本运算符(包括算术运算符、位运算符等)与赋值运算符组合而成,其作用是首先对某变量进行指定的运算,然后再将运算所得的结果赋给该变量。其形式为:

〈变量〉复合赋值运算符〈表达式〉;

表 1—10 列出了各个复合赋值运算符的功能。

表 1—10 复合赋值运算符的功能

运算符	功 能
+=、-=	加法赋值、减法赋值
*=、/=、%=	乘法赋值、除法赋值、取模赋值
<<=、>>=	左移位赋值、右移位赋值
&=、|=、^=	逻辑与赋值、或赋值、异或赋值

8. 逗号运算符

逗号运算符“,”可以用来将多个表达式组合成一个表达式,其使用形式为:

〈表达式 1〉逗号运算符〈表达式 2〉逗号运算符…〈表达式 n〉;

对于由逗号运算符分隔开的各个表达式,在执行过程中总是从左至右依次计算各个表达式的值,并且整个表达式的值就是最右边一个表达式的值。

9. 条件运算符

条件运算符(? :)有时也被称为三项运算符,它是 C 语言所独有的运算符之一,可以认为是条件语句的简略形式。其形式为:

〈逻辑表达式〉? 〈表达式 1〉:〈表达式 2〉;

该运算符的含义是:首先对逻辑表达式进行判断,当其结果为真

时，整个表达式的值就取〈表达式 1〉的值，否则就取〈表达式 2〉的值来作为整个表达式的值。

10. 指针及地址运算符

指针是变量的内存地址，指针变量的值是另一个目标变量的地址。指针与地址运算符分别是 * 和 &，其运算形式为：

(1)取内容　　〈变量〉= *〈指针变量〉；

(2)取地址　　〈指针变量〉=&〈目标变量〉；

11. sizeof 运算符

sizeof 运算符是一种专用于计算类型、变量以及表达式的字节数目(长度)的运算符。在 C 语言中有许多接近于汇编的特性，经常要求计算地址、位置或变量及表达式占用内存大小等，因此专门设计了 sizeof 运算符，其使用形式为：

sizeof(〈表达式〉)；或 sizeof(〈类型〉)；

12. 强制类型转换运算符

强制类型转换运算符用于把表达式或变量的类型强制转换成指定的类型。其使用形式为：

(〈类型〉)〈表达式〉；

这里〈类型〉可以是 §1—3 中列出的所有数据类型。

13. 运算符的优先级与结合规则

在 C 语言中，运算符的优先级与结合规则直接关系到表达式中各个变量之间的组合与计算顺序，从而最终会影响到整个表达式的值。在一个表达式中，与优先级较高的运算符直接关联的子表达式将首先被计算。另一方面，所谓结合规则是指在优先级相同的运算符被并列使用时，决定运算符是从左往右还是从右往左执行的一种规则。在表 1—11 中给出了 C 语言中所有运算符的相对优