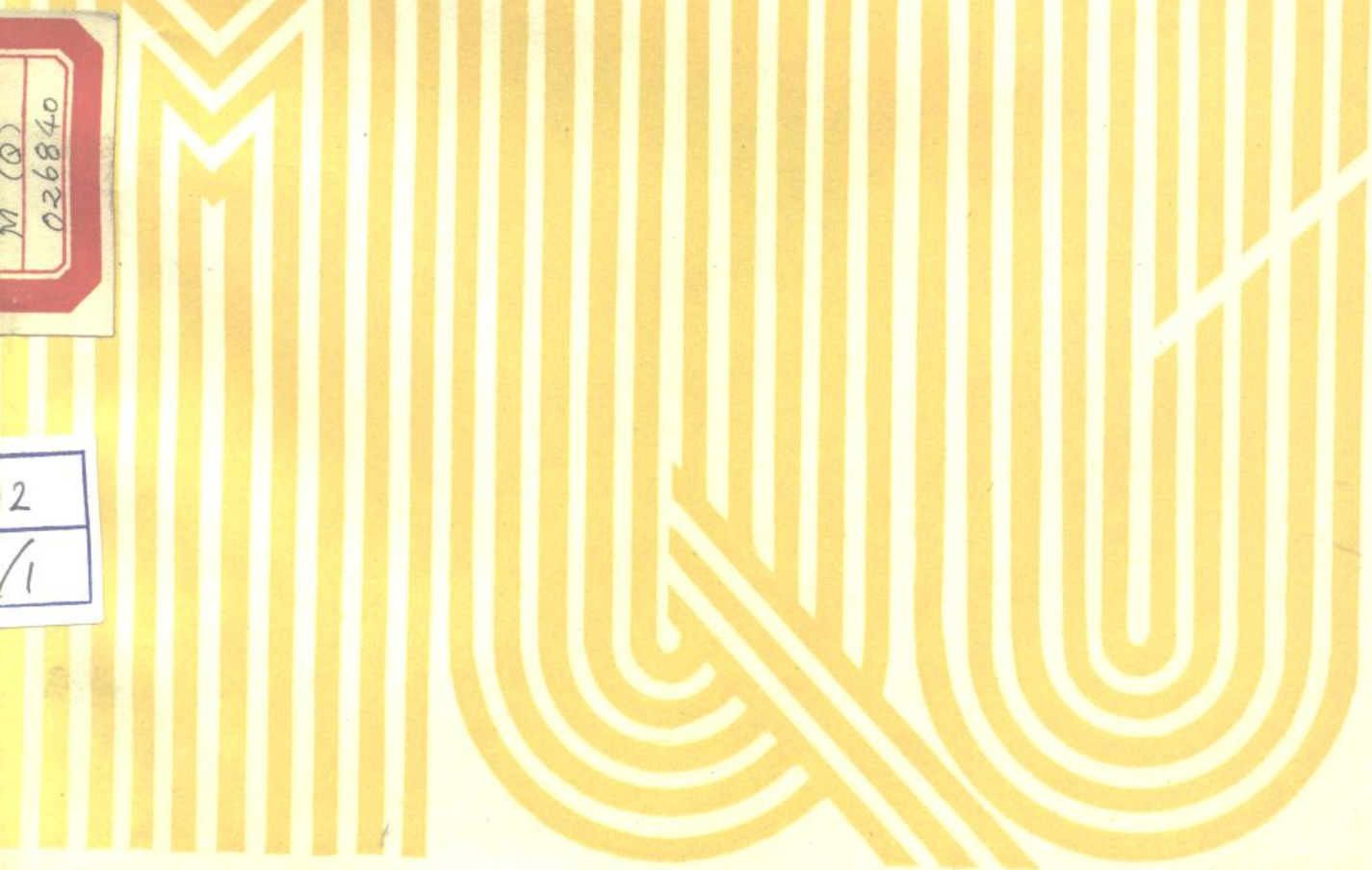
中国科学院计算所公司计算机技术丛书

Microsoft Quick C
语言参考手册

邱 平　韦 帷　编译　　许志平　审校

国防工业出版社



中国科学院计算所公司计算机技术丛书

Microsoft Quick C
语 言 参 考 手 册

邱 平 韦 帷 编译
许志平 审校

国防工业出版社

内 容 简 介

这套丛书共三册，包括《Microsoft Quick C 语言参考手册》、《Microsoft Quick C 程序员参考手册》和《Microsoft Quick C 库程序参考手册》。本书是其中第一册。

Microsoft的Quick C是MS-DOS下的C语言编译器，它代表C语言的最新发展趋势。Quick C集编辑、编译、连接和调试于一体，构成完整的C语言开发环境。Quick C 的特点如下：(1) 它与MSC5.0完全一致；(2) 采用类似Wordstar的编辑方法；(3) 新的编译技术使编译速度达每分钟上万行；(4) 内部MAKE功能允许开发大型而复杂的程序；(5) 内部调试程序允许断点设置，逐行执行等功能；(6) 带有图形功能。

本书主要描述Microsoft关于C语言的实现约定。用户可以从中查到Quick C 的具体约定以及与标准C的差别。它给使用Quick C 的用户提供了必须具备的基本知识。

本书可作为大中专学生在微机上学习C 语言的指导书，对使用C语言在MS-DOS下开发软件的程序员也是一必备的工具。

*

中国科学院计算所公司计算机技术丛书

Microsoft Quick C 语言参考手册

邱 平　韦 帷 编译

许志平　审校

*

国防工业出版社出版、发行

（北京市车公庄西路老虎庙七号）

新华书店经售

北京昌平兴华印刷厂印装

*

787×1092　1/16　印张8 3/4　200千字

1988年12月第一版　1988年12月第一次印刷　印数：00,001—5,000册

ISBN　7-118-00467-7/TP·57

定价：5.80元

编 译 者 序

读者所见的这套手册，是我们根据Microsoft公司最近推出Microsoft Quick C 1.0的一套手册编译的。从我们获得这套软件后，经过了几个月的使用，认为这是一套值得推广的好软件，应该介绍给国内的计算机工作者及大中专学生。为此目的，我们组织编译了这套资料，奉献给大家。

在国内，C语言并不像BASIC语言那样有广大的用户，它主要由大中专学生，软件开发人员和高级程序员使用。到目前为止，国内流行的C语言编译程序为数不少，但它们之间有什么联系和差别呢，如何确认哪种是最适合你的呢？

一、流行在MS-DOS下的C语言

由于工作关系，我接触了若干种C语言编译程序。这里，不打算讨论C语言的整体优点，而只想分析一下这些不同厂家的产品的差异。

从IBM PC系列机一出现，国内市场就出现了Computer Innovation公司的C 86(或者优化C86）编译程序，语言本身可以在CP/M86和MS-DOS下运行，但在当时的环境下，并没有多少人使用（更多人使用BASIC语言，甚至汇编语言）。

同时出现的另一版本是Digital Research公司的DRC语言编译器，由于是从其它机种移植过来的，所以带有原来的痕迹，国内也有一些使用者。

不过，引起大家注意的是Lattice公司的LC语言，从2.12版起，一直到3.0版，我们都用得很多。这是一个较为稳定的编译器，也是DBASE等程序的生成语言（国外有很多用LC编制的应用软件）。它一直引起我们的兴趣，直到Microsoft公司推出了MSC4.0之后，我们才转移了注意力。

实际上，MSC的2.1版我们早已见过，但当时它并不起眼，还不如LC名声大，而且库函数也比LC少。但到了MSC4.0就不同了：它附带了一个调试程序CODEVIEW1.0。这一下吸引了用腻了DEBUG和SYMDEB的程序员。CV1.0是支持源程序级调试的软件，它允许在源程序行中加入断点和逐行执行。由于CV1.0只支持MSC4.0，所以当时都开始使用MSC4.0。

几乎同时，Turbo公司席卷了软件界，Turbo C以它的高速编译吸引了很多人，它每分钟可编译上万行。不过由于它的兼容性较差，所以很多人不敢过分使用。但灵活的小程序还是可以用TC完成。

总的来说，LC、MSC和TC是目前三种最好的C语言编译，它们分别具有以下特征：

LC：可靠性好；库程序丰富。

MSC：与其它语言兼容性好；有CV这样的调试工具，使调试不再枯燥。

TC：编译速度奇快，开发程序容易。

是否有一种集三种语言优点的好编译器呢？目前的Quick C做到了这一点。

二、Microsoft的Quick C 1.0

今年初，我们看到了MSC5.0软件，它比4.0增加了图形功能和简化了与操作系统的接口。另外，在优化方面也做了很多工作。但我们被同时发行的Microsoft Quick C 1.0所吸

引了。它兼有前面三种语言的优点：在库程序方面不少于LC，在语言能力方面等于MSC5.0，并可以用CV2.0调试；它本身提供了一个集成式程序开发环境，在编译速度上可以与TC相比。另外，它还有许多自己的特点。

下面我们详细介绍一下MS Quick C的优点：

·与MSC5.0能力相当

Quick C在编程能力上相当于MSC5.0，它们使用相同的库程序和相同的表达方法。它的源程序可以不加改动地由MSC5.0编译通过。

Quick C有四种不同的模式，可以适应各种具体需要，编出高效且紧凑的代码。它也支持用CODEVIEW调试程序进行调试。

Quick C简化了与操作系统的接口，比起MSC4.0来，与低级系统调用打交道更为方便，形式更加直观。

Quick C增加了图形功能，提供了简单的图素形成手段，充分利用了IBM PC系列机的图形能力。它支持CGA、EGA和VGA多种图形适配器。

·与Turbo C能力相当

Quick C采用了类似TC的集成式程序开发环境，集源程序编辑、编译、连接和调试于一体，构成一个完整的C语言开发环境。

Quick C采用了类似Wordstar的内部编辑程序，便于用户输入和修改源程序。同时提供了自动括号匹配能力，使程序员易于查出括号匹配的语法错。

Quick C利用了内含程序库的技术，使编译速度达到每分钟超过10000行的高速度。在编译过程中，每次可以找出26个错误，编译完成后，把光标停在出错的行上，等待你修改。

Quick C在编译正确后自动进行连接和执行，所以只用按一个键就能让程序自动地通过编译、连接和执行全过程。

·Quick C特有的优点

Quick C包括丰富的库程序。基本库程序常驻内存，支持快速连接。其它库程序和用户自定义库程序可以通过QLIB实用程序构成QLB常驻内存库，使编译速度得到保证。

Quick C支持多模块编译。内部Program List允许定义多个分立模块，从而连接成一个大型程序。

Quick C编译器可以生成内存形式文件，OBJ型文件和EXE型文件，它还带有一个与XENIX兼容的命令行编译器QCL。

QCL支持四种不同的模式，即S、M、C和L模式。它自动调用LINK程序，使编译、连接一气呵成。

Quick C内部含有一个MAKE程序，自动维持程序版本的更新，用户在使用它维护多模块时，完全不必自己管理，Quick C将自动完成。

最令人推崇的特性就是Quick C有内含的调试功能。当编译和连接完成后，Quick C自动处于源程序调试状态，这时，你可以控制程序逐行执行，或执行到指定行，也可以检查变量的值。它很象CODEVIEW，但又比CODEVIEW方便灵活。

三、Quick C的应用

上面我们介绍了Quick C的一系列优点，它们可以在实践中予以检验。我们觉得，这个软件由于具有强大的能力，可以用在以下几个方面。

·用于教学：目前用来教学最多的是Turbo C，原因是它可以快速编制小程序，快速定位错误。但Quick C比Turbo C做得更好。一是查错方便，二是自动进入调试状态，可以逐行执行并检查变量，对于理解某些语句的作用很有帮助。三是这种语言与C语言标准保持一致，既适应ANSI标准，也适应XENIX和MS-DOS，容易正规化。

·用于开发软件：开发软件时三分是编程序，七分是调试程序，有好的调试工具就会如虎添翼。过去曾用过DEBUG和SYMDEM，后来又用过CODEVIEW，越来越高级，但毕竟要费点事来进行转换。现在Quick C内部提供了这个能力，调试时就省了力气，既不必滥加打印语句，也不用耽心死循环，加之还有Mouse的支持，所以“Quick”一词恰如其分。

正因为如此，我们才热心向大家推荐。读者在程序开发上能有更好的前景，我们也就达到了目的。

许志平

1988年6月

目　录

第一章 概 述

1.1 C语言综述

C语言是一种通用程序设计语言，以其效率、经济性及可移植性而闻名。正因为具备这些优点，它能够承担各种程序设计工作，尤其是在系统程序设计中，该语言十分有用。因为程序员可以编写快速而紧凑的C代码，并使之与其它系统程序有良好的界面，在多数情况下，一个好的C语言程序在速度上可与汇编程序相媲美。C语言还具有易于维护和可读性强的优点。

C语言以比较小的语言规模，实现了效率和功能上的统一。C没有内部函数作输入输出、存储分配、屏幕操作和进程控制的工作，但仍可用C语言来完成所需的工作，因为C程序员可依赖程序库来完成这些任务。

这种设计使C具有灵活性和简洁性。它的语言相当节省，它既不假定也不强制规定一种特定的编程模式。该语言提供的程序库对用户的特殊要求提供了支持。如果必要，程序员可对程序库进行适当剪裁再使用。这些设计亦有助于孤立语言的性质，从而帮助程序员写出可移植的代码。C语言的严格定义使之不依赖于任何特殊的操作系统或机器，程序员可以方便地加入指定的系统子程序以提高对机器的使用效率。

C语言具有下面的重要特点：

- 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，这有利于结构化程序设计。
- C语言提供了一个较大的操作符集合。大多数C的操作符是与一般机器指令相一致的，可直接翻译成机器代码，其它操作符明确指定不同的操作，可产生最短的机器代码。
- C语言的数据类型包括几种尺寸的整数以及单、双精度的浮点数，程序员亦可设计更为复杂的数据类型，如数组和结构来适应特殊的程序需求。
- C语言允许程序员定义变量指针和函数指针。指针就是与机器地址相关的说明项。正确地使用指针可提高程序的效率，因为指针是让程序员以相同于机器码的形式存取内存的数据。C语言中还支持指针的运算，允许程序员直接访问和操纵内存地址。
- C语言的预处理亦是一种正文处理，是编译之前对正文文件的再安排。其中，用得最多的是定义程序的常数、代替函数调用的宏（可较快速运行）和基于某种特定条件的编译指令。

C语言是一种很灵活的语言，允许程序员做出各种决定。为了保持这种特性，C语言很少在类型转换等方面加以强制性的限制，这通常是很有益的，但是，程序员必须了解语言特征，以便更好地理解C程序。

1.2 关于本手册的介绍

《Microsoft Quick C语言参考手册》(即本手册)定义了由Microsoft公司实现的C语言。该手册是提供给对C或其它程序设计语言有一定应用经验的程序员的一本参考资料，这里假定读者已具备程序设计的基础知识。

读者如果想快速浏览Microsoft C的明确定义，以及了解与Brian W.Kernighan和Dennis M.Ritchie所定义的C语言的差异，可参看本手册的附录A和B，它们分别提供了本语言的语法和差异说明。

库函数可用于Microsoft C程序中，对它的讨论在另一本《Microsoft Quick C库程序参考手册》中有专门介绍。

《Microsoft Quick C程序员参考手册》说明了如何在你的系统上进行编译和连接，还解释了在系统上实现C程序的特定信息。

本手册的结构如下：

第二章“C的成分”描述了可用于C程序的字母、数字和符号以及对C编译器具有特定含义的字符组合。

第三章“程序结构”介绍了C程序的各种成分和结构，并解释了怎样组织C的源程序文件。

第四章“说明”描述了如何在C语言中对变量、函数和各种用户类型说明其属性。C语言中提供了一些预定义的数据类型，并允许程序员说明复合类型及指针。一种C语言的新特征，函数定义模式亦在这一章和第七章中进行了讨论。

第五章“表达式和赋值”描述了C语言的表达式和赋值的各种操作数及操作符。本章还讨论了类型的转换和表达式运算可能出现的副作用。

第六章“语句”阐述了C程序执行时各种语句构成的控制流。

第七章“函数”讨论了C函数的性质，特别解释了怎样定义、说明和调用函数，并描述了函数的形参使用及返回值等。

第八章“预处理命令及杂注”描述了C预处理器所能识别的命令。C语言的预处理是在编译前自动地对正文进行的一种处理。本章还介绍了可以放在源程序文件中的对编译器的指示杂注。

附录A“差异”列出了Microsoft C和W.Kernighan &M.Ritchic所定义的C之间的差别。

附录B“语法综述”列出了Microsoft C语言的语法规则。

第二章　C语言的成分

2.1　简介

本章描述了C语言的各种成分，包括用来构造C语言程序的各种名字、数字和字符。下面各节将分别讨论下述内容：

- 字符集
- 常数
- 标识符
- 关键字
- 注释
- 单词（语法单元）

2.2　字符集

在C语言中，定义了两个字符集可供使用，它们是“C字符集”和“可表示字符集”。

C字符集由字母、数字和在C语言中具有特定含义的标点符号组成。构造C程序时，可用C字符集中的字符进行适当组合写出具有完整意义的语句，再由语句组成程序。

C字符集是“可表示字符集”的一个子集。可表示字符集包括所有的字母、数字以及可用单字符绘出的任何符号。可表示字符集的范围取决于终端、控制台和字符发生器的类型。

构成C程序的所有字符都必须是C字符集中的字符，但字符串、字符常数、注释和#include命令中的文件名可以例外，它们可以由可表示字符集中的任何字符组成。

C字符集中的每个字符对C编译器来说都有明确的含义，当编译器碰到非法字符或不属于C字符集中的字符时，就会给出出错信息。

2.2.1～2.2.5将介绍C字符集中的各种字符和符号，并说明怎样及何时使用它们。

2.2.1　字母、数字和下底线符号

C字符集包括英文符号表中的所有大写、小写字母、十进制阿拉伯数字系统和下底线符号。

- 大写英文字母：
  ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 小写英文字母：
  abcdefghijklmnopqrstuvwxyz
- 十进制数字：
  0 1 2 3 4 5 6 7 8 9

· 下底线符号：

(_)

这些字母和数字可用来构造各种常数、标识符和关键字，这在本章的后面还要提到。

C编译器把大写和小写的字母视为不同的字符。例如，如果a被指定为某一项，则不能用A来表示该项，而必须使用小写形式。

2.2.2 空白字符

空格符、制表符、换行符、回车符、换页符、垂直制表符都称为空白符，因为它们都是用来在打印页上生成字与字之间、行与行之间的空格的。这些字符将你所定义的项，如常数和标识符，同程序中的其它项分开。

C编译器将CTRL+Z字符作为文件结束识别标志，该标志后的任何正文均被忽略。

C编译器一般忽略空白字符，除非它们用作分隔符或是作为字符常数或文字串的一部分，因此可以在程序中适当加些空白符以增加其可读性，编译器也将注释的内容视为空白符(见2.6节)。

2.2.3 标点符号和特殊字符

C字符集中的标点符号和特殊字符有各种用途，它可以组织程序的正文，也可以定义编译器或被编程序执行的任务。下面的表2.1列出了C字符集中的标点符号和特殊字符。

表2.1 标点符号和特殊字符表

字 符	名 称	字 符	名 称
,	逗号	!	感叹号
.	圆点	|	竖线
;	分号	/	斜线
:	冒号	\	反斜线
?	问号	~	波折号
'	单引号	+	加号
"	双引号	#	井号
(	左圆括号	%	百分号
)	右圆括号	&	and符
[	左方括号	^	脱字符
]	右方括号	*	星号
{	左花括号	-	减号
}	右花括号	=	等号
<	左尖括号	>	右尖括号

这些字符在C语言中都有特定的含义。本册各章节都会涉及它们的使用。表2.1中未列出的可表示字符集中的其它字符，只能用于字符串、字符常数、注释和#include命令的文件名中。

2.2.4 转义序列

转义序列又称为换码序列，是由指定的符号组合表示的一种空白或不能打印的符号，常用于字符串和字符常数中，以表示某种特定的活动。例如，回车和制表跳格等在终端和打印机上的动作，就需要使用转义序列控制。转义序列是由反斜线（＼）开头，后面跟有字母或数字。表2.2开列了C语言的转义序列：

表2.2 转义序列表

符号序列	名 称	符号序列	名 称
＼n	回车换行	＼a	响铃报警
＼t	水平制表	＼'	单引号
＼u	垂直制表	＼"	双引号
＼b	退格	＼＼	反斜杠
＼r	回车	＼*ddd*	八进制ASCII代码值
＼f	换页	＼x*ddd*	十六进制ASCII代码值

如果反斜线后的字符不包括在上述表中，则反斜线不起作用，后面字符仅仅表示字符本身的含义。例如，＼c在一个字符串或字符常数中只表示可显示字符c。转义序列只使用小写字母，这是ANSI为使特征标准化而作出的规定。未定义的转义序列的出现，尽管当前无害，却可能造成将来移植上的困难。

序列＼*ddd*可将ASCII字符集中的任何字符定义为3位八进制字符代码，类似地，序列＼x*ddd*可将ASCII字符集中的任何字符定义为3位十六进制字符代码。例如，可以用正常的C转义序列（＼b）表示ASCII字符集中的退格符，也可用代码＼010（八进制）或＼x008（十六进制）来表示它。

在一八进制转义序列中只能使用数字0～7。在转义序列中至少要出现一位数字，即可用少于3位的八进制或十六进制的代码来表示ASCII字符集中的某个字符。例如，可以将ASCII字符集中的退格符定义为八进制数字的转义序列＼10或是将其定义为十六进制数字的转义序列＼x08或＼x8。

注意：当在字符串中使用八进制或十六进制转义序列时，最保险的做法还是给出全数位的转义序列（3位八进制数或3位十六进制数），否则，紧跟在转义序列后面的字符可能被编译器误认为是转义序列的一部分。例如，如果打印串“＼x07Bell”，将显示{ell，因为／x07B被译为ASCII码字符集中的左花括号字符。而串＼x007Bell将产生正确的显示结果Bell。

转义序列可将不可显示的控制符送到一个显示设备。例如，转义字符＼033常被用作终端或打印机上控制命令的首字符。某些转义序列是设备指定专用的，例如，垂直制表符和换页符（＼v和＼f）对屏幕输出没有任何影响，但对打印机却执行相应的操作。

切记：在C程序中，不可显示的字符必须用转义序列来表示，因为直接使用该字符可能产生编译诊断信息。

反斜线（＼）也可用作续行标志符。当一个回车换行符紧跟在反斜线后面时，编译器就

会忽略反斜线和回车换行符，而将下一行作为当前行的一部分。当一个预处理定义大于一行时，就可以这样做。

2.2.5. 操作符

操作符（单个字符或字符组合）是对操作对象指定动作的一些特殊符号。编译器将这些符号中的每一个解释为相应的程序元素，称之为特殊“单词”（见2.7节）。

表2.3中列出了C语言的一元操作符及每个操作符的名字。表2.4中列出了C语言的二元和三元操作符及其名字。在使用中必须按照表中开列的形式正确地书写，在多字符的复合操作符中间不得使用空格。注意有三个操作符（★，－，&）在两个表中都出现了，但含义并不相同，要根据上下文确定它们到底是一元操作符还是二元操作符。另外，sizeof操作符不包括在这个表中，因为它是以关键字的形式出现的，并不是特殊字符，所以被列在2.5节中。

表2.3 一元操作符

操作符	名字	操作符	名字
!	逻辑非	*	间接
~	按位求补	&	取地址
—	算术负	+	一元加

表2.4 二元或三元操作符

操作符	名字	操作符	名字
+	加	&&	逻辑与
−	减	! !	逻辑或
★	乘	,	顺序求值
/	除	? :	条件①
%	取余	++	增量
<<	左移	−−	减量
>>	右移	=	简单赋值
<	小于	+=	加赋值
<=	小于等于	−=	减赋值
>	大于	★=	乘赋值
>=	大于等于	/=	除赋值
==	等于	%=	取余赋值
! =	不等于	>>=	右移赋值
&	按位与(and)	<<=	左移赋值
!	按位或	&=	按位与赋值
∧	按位异或	∧=	按位异或赋值
! =	按位或赋值		

① 条件操作符是一个三元操作符，而不是一个两字符在一起的操作符，在问号和冒号之间还有一条件表达式，其形式如：

表达式？表达式：表达式