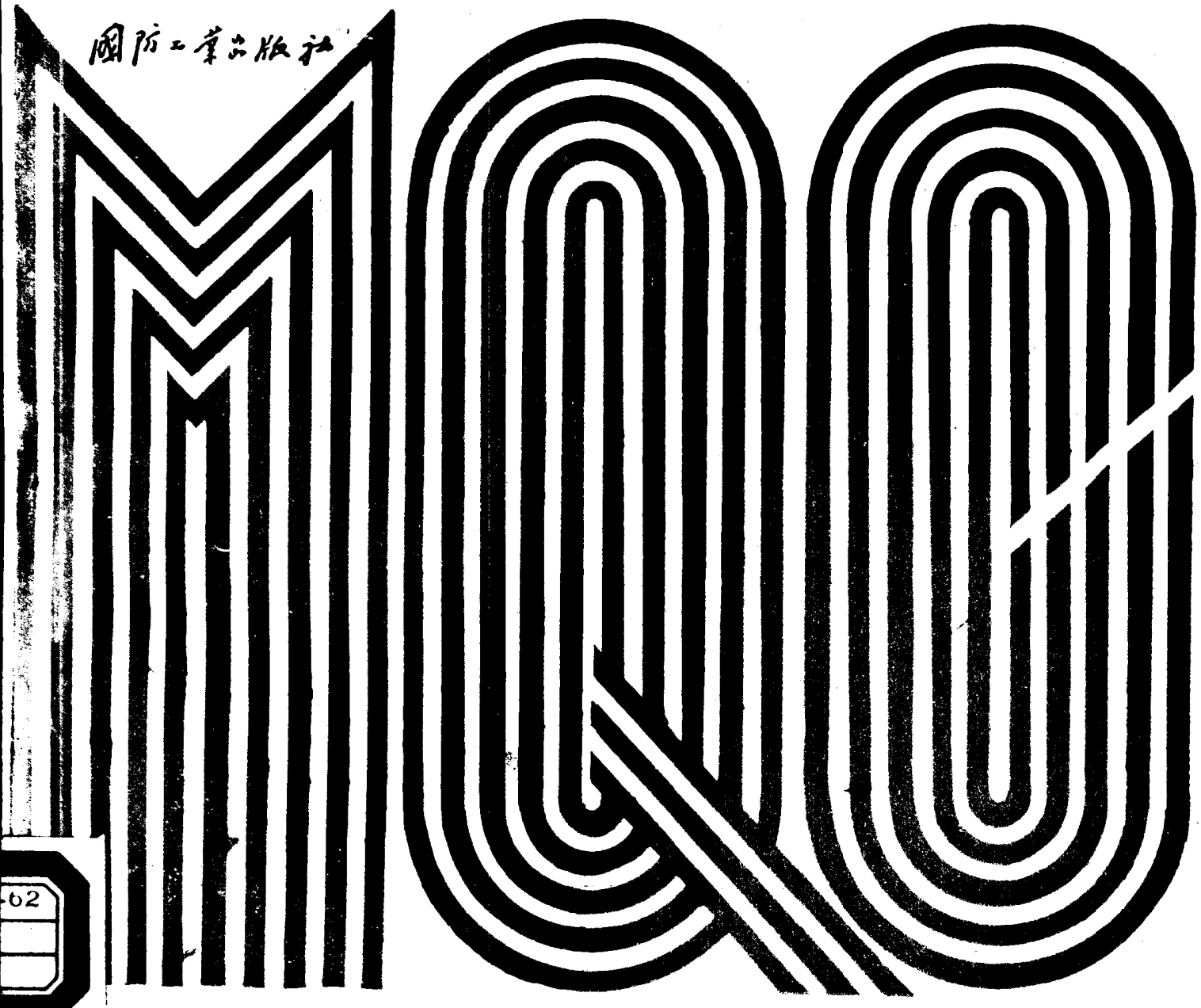
中国科学院计算所公司计算机技术丛书

Microsoft Quick C
库程序参考手册

吴 双 章立生 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

库程序参考手册

吴 双 章立生 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》。本书是其中第三册。

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和逐行执行等功能；(6) 带有图形功能。

本书首先概括介绍了Quick C 程序库中的过程和变量的用法及分类，然后介绍了include 文件的各种定义。最后按字母顺序对每个库函数的语法、语义、范例和注意事项给出详尽的说明。

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 库程序参考手册
吴 双 章立生 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22 3/4 517千字
1988年12月第一版 1988年12月第一次印刷 印数：00,001—5,000册
ISBN 7-118-00470-7/TP·59 定价：10.75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，我们接触了若干种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IBMPC系列机一出现，国内市场就出现了Computer Innovation公司的C86（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语言编译器，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MSC 5.0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MSC 5.0，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MSC 5.0能力相当

Quick C 在编程能力上相当于MSC 5.0，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MSC 5.0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。基本序程序常驻内存，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。它很像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，这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Ouick 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

第一部分　概　述

第二部分 参考手册

附　录

第一部分　概　　述

第一章 引 言

1.1 关于C程序库

Microsoft C程序库 (The Microsoft C Run-Time Library)包括了用于C语言程序设计中的200多个预先定义的函数和宏。C程序库通过提供以下功能，使程序设计更加容易：

1.与操作系统功能（如打开文件和关闭文件）的接口；

2.实现程序设计常用任务的快速高效的函数（例如字符串处理），这节省了程序员用于书写这些函数的时间和精力。

因为C语言本身不提供输入、输出、存储分配和进程控制等基本函数，所以使用C语言的程序员必须依靠程序库来提供这些函数。因此，在用C语言进行程序设计时，C程序库就显得特别重要。

在设计Microsoft C程序库的函数时，保证了它们与MS-DOS和XENIX或UNIX系统之间最大程度的兼容性。在这本手册中，凡涉及到XENIX的内容，对UNIX和其它类UNIX系统也同样适用。

在MS-DOS中的C程序库中，大多数函数同XENIX中C程序库同名的函数在功能上是兼容的。如果你对可移植性感兴趣，请参阅附录B。这个附录列出了C程序库中对MS-DOS特有的函数，并且叙述了MS-DOS和XENIX系统中C程序库相同名函数之间在功能上的差异（如果有差异的话）。

为进一步提高兼容性，Microsoft C程序库中的数学函数经扩充后都具有异常情况处理功能，其处理方式与UNIX系统V中的数学函数相同。

除了那些国际通用化的函数外，该程序库也设计成与美国国家标准草案（ANSI C）相兼容。与ANSI C标准相一致的函数也在附录B中列出。

为方便那些有兴趣使用MS-DOS特殊功能的程序员，C程序库包括了MS-DOS接口函数。这些函数允许从一个C语言程序中使用MS-DOS的系统调用和中断。C程序库中也包含控制台输入输出函数，以便从用户控制台上有效地进行读和写。

为了利用Microsoft C编译程序的类型检查功能，对伴随C程序库的include文件进行了扩充。除了库中的函数和宏所需要的定义和说明外，include文件中还包含带有形式参数类型表的函数说明。类型表使系统能够对库函数的调用进行类型检查。这个特点有助于查出由于函数的形参与实参类型不一致造成的程序错误。

为了给程序库中的函数提供参数类型表，在C程序库中的标准include文件表中增加了几个新的include文件。这些新的include文件的命名做到了尽量与ANSI C标准草案以及XENIX和UNIX中的名称一致。

1.2 关于本手册

本参考手册描述Microsoft C程序库的具体内容。本手册假定你熟悉C语言和MS-DOS，假定你知道怎样在你的MS-DOS系统中编译和连接C程序，并且能够使用环境变量建立编

译器和连接程序环境。如果你对编译、连接或建立环境有疑问，请参阅《程序员参考手册》；如果你不熟悉C语言，请参阅《语言参考手册》。

注意，因为MS-DOS和PC-DOS是本质相同的两个操作系统，因此如果差别不大，本手册均采用MS-DOS代表这两个操作系统。

本手册划分成两大部分。第一部分是概述，介绍C程序库，它所讨论的是适用于整个库的一般规则，并概括了程序库的各成分。

第二部分是参考手册，它按英文字母的顺序叙述库中的函数，以供快速查阅。一旦你熟悉了库的规则和过程，你就会经常使用本手册的第二部分。

第一部分其余各章的内容是：

第二章（使用C程序库）给出了理解和使用C程序库的一般规则，指出了适用于某些库函数的特殊要求。建议你在使用程序库前阅读这一章；当你对库过程有疑问时，也可查阅这一章。

第三章（全局变量和标准类型）叙述在程序库中定义并为程序库函数使用的变量和类型。这一章还提供了对定义（或说明）这些变量和类型的include文件的相互参照表。你会发现这些变量和类型在你的程序中也是有用的。这些变量和类型在后面各章节的相应部分也有所描述。

第四章（库程序分类）对库中的函数进行分类，列出每一类中的诸函数，并讨论每一类的特殊性。本章是第二部分的补充，使用户容易按照功能寻找要用的库程序。一旦找到了你所需要的库程序，就可从相应的参考部分获得详尽的内容。

第五章（INCLUDE文件）概括了与程序库一起提供的每个include文件的内容。

第二部分后的附录提供了有关出错信息和与XENIX兼容的库程序的更详尽的资料。附录A（出错信息）描述了使用库程序时可能出现的错误值和信息。附录B（通用库）列出了MS-DOS C程序库中与XENIX和UNIX系统中同名的库程序，以及与ANSI C标准草案库程序兼容的库程序。附录B中还讨论了这些库程序在DOS和XENIX系统中的差异（如果有差异的话），并讨论了通用全局变量和include文件。

第二章　使用C程序库

2.1　引言

为了使用C程序库函数，只需在你的程序中直接调用库中的例程，就象它们是在你的程序中定义的那样。C程序库函数是以编译后的形式存放在库文件中的。

在连接时，你的程序必须与相应的一个或多个文件连接在一起，以便解决对库函数的引用和为被调用的库函数提供代码。与C程序库的连接过程在《程序员参考手册》中详细讨论。

在大多数情况下，你必须通过履行下面两个步骤(或其中之一)为调用C程序库函数作准备：

1.在你的程序中嵌入一个指定的“.h”文件。许多库函数都需要由某个include文件提供的定义和说明。

2.为那些返回值的类型不是int的库函数提供说明。因为如果未说明函数返回值的类型，C编译器就认为它返回值的类型为int。你可以通过在程序中嵌入包含这些说明的C库文件或者通过显示地说明函数的方法来提供这些说明。

这些是使用C程序库函数的起码要求；你或许还得执行一些其它步骤，如使编译器实现对函数调用进行参数类型检查。

本章其余部分讨论使用程序库函数的准备过程和一些对某些库函数适用的特殊规则（如文件名和路径名的约定）。

2.2　区分函数和宏

在本手册中，“函数”和“例程”是作为同义词使用的。事实上，C程序库中大多数例程都是C函数，即它们是由编译过的C语句组成的。但是，有些例程是作为“宏”实现的。一个宏是由预处理命令#define定义的标识符，它代表一个值或一个表达式。和函数一样，宏定义中也可以带有零个或多个形参。定义和使用宏在《语言参考手册》的第八章中有详细讨论。

在C程序库中宏的定义与函数的作用一样：它们也带参数并返回值，而且调用方式也与函数相似。使用宏的主要优点是运行速度快；它们的定义在预处理阶段就被展开，省去了函数调用所需的开销。但是，由于宏要在编译前被展开（即用其定义去替换），故它们会使程序的代码长度增大，宏在程序中出现的次数较多时更是如此。函数无论被调用多少次，它们在程序中只被定义一次；相反，宏的每一次出现都要被展开。因此，函数和宏提供了对程序的运行速度和代码长度之间的一种权衡机会。有时，C程序库为某例程同时提供了函数和宏两种实现方式，使你可以从中进行选择。

下面列出了函数和宏之间的一些重要区别：

1.当在宏定义中对具有副作用的参数的计算多于一次时，有些宏对这些参数的处理可能