北京希望电脑公司C语言技术丛书

软件开发人员必备工具书

Microsoft C 6.0

经　典

章含　编

海洋出版社

北京希望电脑公司C语言技术丛书

软件开发人员必备工具书

Microsoft C 6.0 经典

☆ 兼容MSC_3 MSC_4 MSC_5 QC TC ANSI UNIX V

☆ 举例详解500多个MSC库函数

章 含 编译

海洋出版社

1992.北京

内 容 提 要

Microsoft C6.0是C编译器的最佳版本之一，本书分五部分，共19章。

本书提供500多个Microsoft C库函数，以及有关编程结果的信息．书中包含每个例程的实际举例，而每节前都有一个导引，以帮助读者快速而有效地掌握C代码。此外，每个函数中都有一个唯一可兼容框，表明该函数与其它C版本和通用操作系统的可兼容性，更易于读者进一步掌握每个函数。

本书适用于广大计算机用户以及软件开发人员，同时可作为大专院校师生必备的参考用书。

需要本书的用户可直接与北京8721信箱联系，邮编100080，电话：2562329。

(京)新登字087号

责任编辑 阎世尊

Microsoft C 6.0 经典

章含 编译

希望 审校

*

海洋出版社出版（北京市复兴门外大街1号）

海洋出版社发行 北京海淀东升印刷厂印刷

开本：787×1092 1/16 印张：42.875 字数：940千字

1992年2月第一版 1992年2月第一次印刷

印数：1—3000册

*

ISBN 7-5027-2728-0/TP·103

定价 31.00元

前　言

Microsoft C是大型软件系统。目前尚无一本完整地描述其特性、工具、辅助、扩充和细则的参考书。Microsoft C最大价值之一是其庞大的C函数调用库。Microsoft C库拥有500多个例程，可完成任何一项任务，包括从计算正弦，画像素表，到执行串数组的快速排序。

不过，其巨大的潜能还尚未挖掘出来。由某些厂家提供的手册是如此泛泛而谈，以至很难将其用于参考，更无法满足学习的需要。而我们编写的这本书，是从程序员使用方便的角度出发的，书中不仅溶进了程序员在工作中所需的信息，而且还提供了掌握C库所需的背景资料、语法和实例等。

坦率地讲，编写本书的初衷是因为我们在编写一个大型C程序时手头需要这样一本书。因而，本书适用于MS-DOS和UNIX环境，有助于我们查找函数，并为此提供了清晰而简明的导引。

本书逐一描述了Microsoft C库中的500多个函数，并为每个函数提供了基于MS-DOS的实际举例。同时，将例程按功能分类，每类前均有一个中间级导引，后面则是按字母顺序排列的引用项。此外本书还具有以下特点：

△ 两个快速开端导引：一个是有关C语言基本元素的，而另一个是有关运行Microsoft C编译程序的，并包括其主要特性和选项。

△ 详解每类函数是如何相关的，如何挑选正确的函数完成给定的作业，以及有关的注意事项。

△ 本书所提供的实例要比其它手册中提供的举例更为广泛且更为有趣。

△ 提供对读者有益的、且特定于MS-DOS的信息。

△ 提供一个检查框，使读者可清楚地了解给定的函数是否与如下内容相兼容：Microsoft C编译器的早期版本，新的ANSI标准，UNIX系统V库，Xenix，Quick C，Turbo C，Turbo C++及DOS和OS/2操作系统。

△ 通过两个表（一个按字母排序，而另一个按功能分类）快速查找任意函数。

△ 对进一步阅读C语言，编程概念和算法等参考资料提出建议。

总之，随着对本书的深入学习，读者的编程能力会逐步提高。对于初次编写程序的读者来说，本书可作为学习蓝本。通过循序渐进地阅读书中导引，读者定会掌握C语言的全部要旨。对初学者来说，可详细地学习和掌握各类函数。

对有经验的非C语言程序员来说，这本书会使其迅速进入C语言的学习。由于这部分人员已了解程序的功能及用途，所以他们很快便可找到与Microsoft C的相关之处。而对于有经验的C程序员来说，则可跳至第二部分，从而在快速掌握编译程序之后找到其中的相关部分。此外，本书还有助于读者通过找到等效函数将应用程序移植到Microsoft C中。

介　绍

一、总体编排

本书分为如下五部分：

1．第一部分：C语言和Microsoft C

这一部分涉及C和Microsoft C的基本特性，适用于中初级C程序员。其中包括：第一章，“C语言概述”。简明扼要地讨论了C编程语言，其中引用了ANSI扩充。如果读者已了解C语言，可跳过这一部分。第二章，“Microsoft C 6.0编译器特性和选项”。讨论了特定于Microsoft C的C编程关键字和特性。例如，这一章描述了Microsoft C所提供的存储模式，并详解编译器CL的命令行选项。

2．第二部分：进程控制和内存管理

这部分包含对Microsoft C库函数的导引和参考页，注重于进程管理，函数间的通信以及内存管理，并描述了四类函数：进程控制，可变长度变元表，内存分配和管理及缓冲区操作。

3．第三部分：数据处理

这部分涉及处理，转换，计算和操作数据的例程。象算术计算，查找和排序之类的任务在此进行了讨论，并包含如下几类函数：数据转换例程，算术例程，字符分类和转换，串比较和操作，查找和排序以及时间。

4．第四部分：文件和I/O

这部分注重于操作文件以及执行输入和输出操作。MS-DOS和BIOS接口例程也在此涉及到。本部分包含如下函数分类：文件操作，目录操作，输入和输出例程以及系统调用。

5．第五部分：图形

本部分的前三章描述在Microsoft C 5.0，5.1和6.0中所提供的图形例程。第十七章，“图形方式，坐标和属性”包含图形编程所需的一切预备信息，其中包括设置颜色，线形式和填充屏蔽，以及制定图形库内部所含参数状态的信息。第十八章，“画图和动画制作”涉及用图形例程所画的对象，其中包括点、线、长方形，椭圆，弧形和像素。此外，还讨论如何执行动画制作，如何使用Microsoft所提供的新呈现图形函数和制图函数。第十九章，“图形和文本”描述了文本输出例程，并指出如何控制文本在屏幕的显示形式，以及如何把文本仅限于某一窗口中。

二、章节编排

从第三章开始，每一章开头都有一个有关待讨论的例程分类导引，每个导引建立理解和使用该类例程所需的概念。在每一类中，例程既按字母编排，也按功能编排。导引说明了同类中的函数是如何相关的，并详细讲解了它们之间的相似与区别，这样读者可知道哪些类似的命名函数适用于某一特定情况，并使用这些函数执行常见的编程任务。

导引后面是该类函数（以字母序编排）的引用项。引用项提供该函数的目的，语法和用法，并包含举例调用和举例程序。

一般来说，同一函数有若干基于所用内存模式，数据类型，或图型坐标系统的变体。例如，通用的内存分配例程malloc作为变体_fmalloc适用于远指针，作为_halloc适用于大 型存储模式，作为_nmalloc适用于近指针，而作为_bmalloc适用于microsoft C 6.0所提供的基指针。由于所有这些函数都有相同之处，所以本书把它们编辑到单一项中，在malloc下按字母排序。不过，本书中的表格列出了所有函数，这样便于迅速找到相关项。

目　录

第一部分　C语言和Microsoft C

第一章　C语言概述

1.1　介绍

C语言已由ANSI标准所定义，并由关键字的最基本内核所组成。其中关键字提供控制结构和数据类型定义，而内核则由输入和输出（I/O），算术计算，串操作和其它类属计算任务的标准函数库所构成。将它们限定于这些标准关键字和库函数的程序可移植到其它ANSI标准编译器中，甚至在一台带有不同处理器和层次结构的机器上运行的程序也可移植，不过，程序员必须注意到某些数据类型大小的不同，以及字节存储到内存中顺序的不同。

除标准函数外，现代C编译程序说明了许多其它函数，以便在特殊的系统结构（如使用Intel80×86处理器的PC可兼容系统）或在特殊的操作系统（如DOS或OS/2）下运行。因此，PC可兼容编译程序的生产厂家和销售单位针对如下内容添加了许多特定于PC的函数：内存分配，访问外部设备，调用DOS和BIOS服务，图形和其它方面。通过访问系统的所有组成部分有助于充分利用系统的潜能。

多少年来，Microsoft C除提供ANSI标准所需的库函数，还提供许多新的库函数，从而不断扩展和增加其对PC可兼容结构的支持。1987年，MSC 5.0问世，它增加了库，特别是在访问DOS和BIOS服务的系统调用方面和改进图形功能（包括对VGA和其它图形标准的支持）方面尤为突出。1988年，MSC5.1问世。它提供对OS/2操作系统的支持。用它可建立与OS/2保护方式和实际方式（功能上等效于DOS3.X和4.X）一同工作的程序。但总体来看，版本5.0到5.1的修改是较小的。

1990年推出的版本6.0添加了150多个函数，其中包括图形呈现例程和制图例程的一个完整库，在支持内部汇编语言指令并在指针和内存模式的处理方面更为灵活。

随着C库功能和编程环境复杂度的不断增加，程序员们也就更加需要有助于管理的工具。为了满足这一需求，Microsoft C 6.0提供了程序员工作台（PWB）软件包，可自动重新编译的NMAKE和增强型CodeView符号调试程序，用这些开发工具，Microsoft C 6.0为程序开发提供了一种最佳环境。

既然ANSI C标准是已即定的，那么，可移植性可得到进一步提高，而编程实践（如使用函数原型）将有助于使程序更易调试和维护。Microsoft C 6.0提供了一种通常符合ANSI C标准的语言实现。它还使编译程序与UNIX系统V3.4相兼容。

这一章主要是对C语言本身作一回顾，并不考虑Microsoft C的特性。第二章将概述特定于Microsoft C 6.0的特性，并描述用于建立程序的编译程序和链接程序。

在下面讨论C基本特性时，我们将指出ANSI C是如何影响到某一特性的。为便于阅读，用〔ANSI〕字样对这些注释加以标记。

1.2　C程序结构

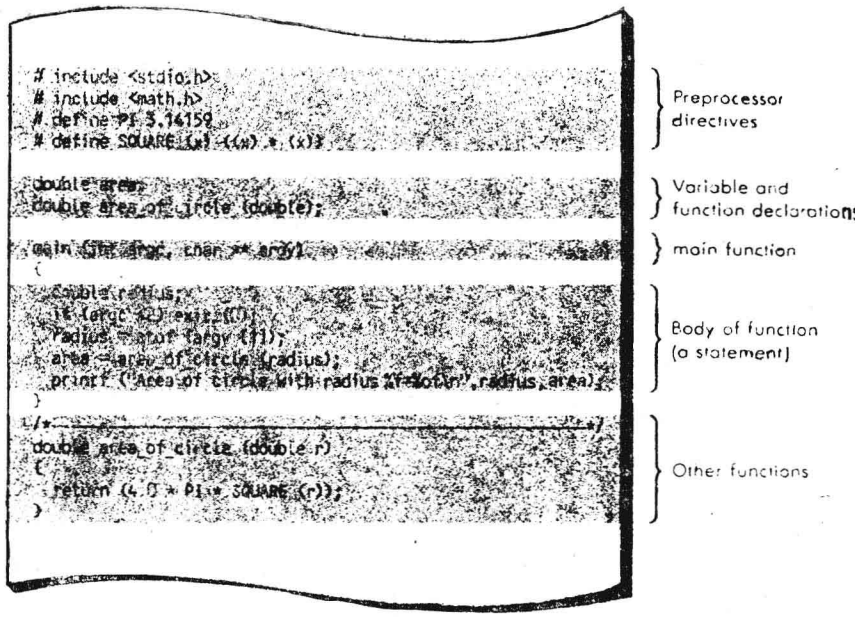

图1-1 C程序结构

正如图1-1所示，典型的C程序由预处理命令（preprocessor directives），变量和函数说明（variable and function declarations），主函数（main function），函数体(body of function）和其它函数（other functions）所组成。每个函数（包括main）体含有局部变量说明，以及表达式和语句。

一个独立的C程序必须含main函数，而不是每个C源文件都要包含一个main函数。在许多应用程序中，可用LINK或类似实用程序将在各个源文件中已测试的函数与主程序链接在一起。这些附加的文件是不会有主函数的，因为main在已编译的C程序中仅出现一次。C程序永远从main的定义开始执行。

1.3 注释

正如例中所示，用符号/*和*/括起来的注释用于解释程序的各个部分。注释可完整地出现在一行，也可分为几行，例如：

```
/* The comment starts here
   and continues here
   and ends here */
```

许多语言允许注释嵌套：一注释在另一注释中。由于编译器忽略注释中的内容，所以代码的某些部分可在构造程序的各个阶段“注释掉”（例如，不需要运行的代码）。C不允许注释嵌套。

从版本5.1开始，Microsoft C允许用一种替换（但不可移植）方法标识注释：用一对反斜线（\\）开头。这就是说，从\\到行尾的内容都是注释。例如：

```
printf ( "CYR Associates" ); \\ company name
```

或

```
printf ( "Production Department Schedule" );
\\ company name
```

在这两种情况下，“\\company name”是标识报告标题的注释，在第二个例子中，注释占用了整个一行。Microsoft的版本5.1和6.0接收任意一种形式，是不会通过第三级报警信息报警的。不过，版本6.0推出了第四级报警，它会在该级标志一个非标准C扩充，因此，允许标识使用\\约定。由于注释约定适用于C++，所以它会在将来流行起来。

1.4 预处理命令

预处理器是唯一于C的功能。在开始编译之前它处理源文件，执行诸如把一个或多个文件的内容合并到C程序（#include命令）中的任务，并通过该程序用一个串模式替换另一个串模式（#define命令）。

预处理器处理程序文件的源文本并操作嵌入到文本中称为预处理命令的命令。这些命令以字符#开头，并约定这样的行不缩排，除非嵌入到其它预处理命令中。在Microsoft C 6.0中有12个预处理命令：#define，#elif，#else，#endif，#error，#if，#ifdef，#ifndef，include，#line，#pragma和#undef。版本6.0还有四个预处理器“操作符”可用在某些命令中：token-pasting，stringizing，charizing和defined。注意，预处理命令并不像实际程序语句一样，不以分号结尾。

在开始编译之前，编译程序调用预处理器，但Microsoft C的最新版本允许运行预处理器并执行指定的替换，但并不实际编译程序。/E选项将预处理结果写到标准输出中并包含#Line命令，/EP选项将处理结果写到标准输出中并去掉#Line命令，/P将预处理结果写到文件中并追加.I扩展名，而/C（仅当与/E，/P或/EP同用时才合法）在预处理过程中保留注释。这些选项对于确保预处理命令获得满意的结果是很有帮助的。

预处理器提供三项重要服务。这三项服务能让用户使程序模块化，更易阅读，且更易移植到不同的计算机中。用#include可把文件内容与C源文件合并。用#define可用一个串替换另一个串(也称为标记替换和宏处理)。用#ifdef类型语句使编译仅限于源文件的选择块（条件编译)。例如，不同的文件可以合并和编译，这基于目标系统是否有数字协处理器，或基于程序是否在DOS或OS/2下运行。

1.4.1 预处理器定义

图1-1中的程序使用了两条#define语句，一个针对常量数值，一个针对串。C程序员常常这样使用，且许多有关C的书都谈及到这一点。但对ANSI C来说，它已过时了。当前人们都使用关键字const（在版本5.1中提供的一种ANSI结构）而不用#define。const关键字将数据项标识为在程序中不能修改的一种常量值，且可与赋值同用，以建立初始化。例如，const float PI=3.14159F与#define PI 3.14159F是一样的。

两条语句在3.14159后面都包含一个“F”，因为没有后缀的浮点常量拥有双精度(double)类型。如果去掉F，那么当PI或类似的定义变量与一般浮点变量进行计算时，编译程序会产生“在浮点类型之间转换”的报警信息(版本5.1的第二级警告，版本6.0的第四级警告)。因此，在TYPICAL.C举例程序中要进行两处修改，以防止接收报警信息：

```
#define PI 3.14159
```

应改为

```
const float PI=3.14159F;
```

而

```
crcmfrnc=2.0 * PI * radius;
```

应改为

```
crcmfrnc=2.0F * PI * radius;
```

同等重要的是，用const可确保编译程序对常量值执行全面的类型检查，例如，确保用于调用函数的值匹配函数参数表中所指定的类型。

1.4.2 文件包含

把其它文件包含在被编译程序中的能力是为模块化而提供的，因为已测试的可靠代码可方便地综合到已开发的程序中。这是特别重要的，因为ANSI使用方法对函数原型进行调用。重复使用的说明可保存在文件中，并包含在需要使用#include的地方。#include语句有三种格式。

```
#include <stdio.h>
#include "local.h"
#include "c:\test\specific.h"
```

第一种格式是最常用的。它告诉预处理器从缺省包含目录中读取命名文件。在Microsoft C中，缺省包含目录是由编译器选项/I directory或由环境变量INCLUDE所指定。尖括号（< >）告知预处理器查看包含目录。查找先在/I开关所指定的目录中开始，之后，如果必要，可在INCLUDE变量指定的目录中查找。我们的举例查找stdio.h文件。该文件是一个标准C头文件。它是Microsoft C版本6.0所提供的40多个头文件中的一个。

第二个例子指定包含“local.h”文件。编译程序认为双引号标记为需要查找源C文件所在的目录。如果包含语句所在文件本身是一个包含文件，那么查找递归地继续下去，直到找到原始C源文件的目录为止。如果对C源目录的查找没有寻找到适当的文件，那么编译程序缺省在上例中所用的查找路径，用/I目录，之后，如果需要，可使用包含语句所用的目录。

第三个例子是最有限制性的。只有待为文件查找的目录是c:\test。一般来说，切记，编译程序不会在任意未用这些规则指定的目录中查找。用/X开关可超越包含查找的标准目录。

INCLUDE环境变量可包含若干目录并引用若干驱动器。如下举例指定三个不同的目录。包含文件可存储在这三个目录中。前两个目录在D驱动器中，而第三个目录在C驱动器中：

```
INCLUDE=d:\MSC\MINC; d:\ZINC; c:\CTUL\TINC
```

1.5 标记替换和宏处理

常见的预处理器任务是用不同的串替换另一个串的所有出现。这是用#define命令进行的，且是字处理查找和替换命令的C等待命令。这一命令常常用来定义数值常量的符号常量，这大大提高了源代码的可读性（正如前面解释的那样，#define的这一用法会导致意想不到的问题，且建议不再这样使用）。#define命令采用这两种形式，其中一种带可选参数，例如：

```
#define PI 3.14159F
#define SQUARE(x)    ((x)*(x))
```

第一条#define命令是一种过时的使用方法，它指定用串3.14159替换串PI在源文件中的每次出现。在开始编译之前，预处理器将查找该串并用指定值取而代之。第二行定义一