Microsoft C 6.0

C语言参考手册

北京科海培训中心

Microsoft C 6.0 之三

C 语言参考手册

北京科海培训中心

前　言

MSC 6.0 是 Microsoft 公司新近推出的 MSC 家族的又一新产品，比起早期的 MSC 产品，MSC 6.0 在很多方面所较大改进。主要体现在:

① 引进程序员工作台的概念，使能开发出运行环境与开发环境不同的应用程序。
② 高度集成化的程序员工作台可以大大加快开发过程。
③ 通过引入集成化的汇编程序，能够在 C 程序中直接使用汇编语言。
④ 最新版本的CodeView 3.0可以在1MB以上的286／386环境下以实方式调试任意大小的程序。
⑤ 功能强大，内容丰富的联机参考系统包括几乎全部有关 MSC 6.0 的信息。
⑥ Source Browser 可以查看程序中各个函数及变量之间的关系。
⑦ 对于 OS／2 应用，增加了动态链接库和多线索与调试选择项。
⑧ 程序员工作台中的编辑器支持鼠标与窗口。

以上新特性使得 MSC 6.0 的功能更强大，界面更友善。

全套 MSC 6.0 技术丛书共有以下四册:

. 《安装及使用手册》介绍MSC 6.0的安装过程以及如何使用程序员工作台和联机帮助系统。
. 《C语言参考手册》介绍MSC 6.0中的PWB、编辑器、编译器、链接器、CodeView 以及其它实用程序的用法。同时介绍 MSC 6.0 的全部库函数。
. 《CodeView及实用程序手册》详细介绍最新版本的CodeView及全部实用程序的使用方法。
. 《高级程序设计技术》介绍一些高级的程序设计技术，诸如优化设计、嵌入汇编、混合语言编程、图形应用等多方面。本书中有大量例子程序。

此外，联机参考系统中还有完整的 MSC 6.0 参考信息。

本套丛书由中科院软件所研究所翻译校对和排版录入。参加本套丛书翻译工作的人员有: 查良钿、孙华、朱力、黄辛卉、庞林、尤国芳、吴倩、何俊、陆强明、白光野等。特别感谢张颖、伍霄音、丁华和李郡，他们为本书的录入校对作了大量工作。

中科院软件研究所
一九九一年一月

目　　录

简　介

一般介绍

Microsoft C 参考手册包括了有关语言，运行库以及 Microsoft C Professional Development System 6.0 实用程序的说明，本书为有经验的程序设计者提供一些特殊的内容，如任选项的含意，编译指令语法以及库函数参数。为方便读者查阅，大量的内容以表格形式列出。

这本书是其它 Microsoft C 资料的补充。象《安装及使用手册》、《高级程序设计技术》和《联机帮助系统》。例如，《安装及使用手册》介绍了 Microsoft C，并教你如何使用程序员工作台（PWB），而本书假设你已经学会使用 PWB，这样本书按照字母顺序和用途列出了所有功能和键命令以及返回值和编辑器开关，但并没有教你如何使用它们。你可以在《高级程序设计技术》一书中找到优化 C 程序的方法。在本参考手册中，你可以找到所有的编译程序任选项说明表，包括那些用于优化的项，以及对每一项的简短说明，但并不针对某一具体情况加以说明。

有关本书的介绍

这一部分将向你介绍本书包括的内容。

1. 实用程序

本手册的第一部分简述了 Microsoft C Professional Development System 包含的实用程序。实用程序按照字母顺序列出。简短的说明和语法在每一实用程序的开始给出，接下来是命令行的任选项。描述中也可能包括其它有用的信息。例如在 LINK 部分列出了在 Windows TM 和 OS／2 应用中用的模块定义文件语句；在 CodeView 调试程序这一项列出了会话命令以及数据大小与格式说明符。

2. 语言参考

第二部分包括了 C 语言的基本内容：列出了所有关键字和转义序列，并给出了语法和语句的简要说明及预处理器命令和编译指令。对操作数和数据类型大小也提供了信息。

3. 运行库函数

Microsoft C 库包括 500 多个函数。第三部分首先分类地列出这些函数，然后按照字

母顺序给出每一函数的描述。列出函数的原型，包含文件，参数表，简要的功能说明和兼容性。

4. 附录

附录部分包括各种有用的表格，例如 printf 和 scanf 格式代码。同时也列出了编译程序和某些程序在运行时的限制，LIMIT.H 中定义了数据范围，FLOAT.H 中定义了算术值。最后还有数值和 ASCII 转换表。

注意：书中所用的“OS／2”是指 OS／2 系统，即 Microsoft Opercrting System／2 (MS OS／2) 和 IBM OS／2，同样“DOS”指的是 MS-DOS 和 IBM Personal Computer DOS 操作系统。如果需要，会指明特殊的操作系统的名字。

第一部份　实用程序

BIND

简述：BIND转换保护式程序，这些程序使用Family API函数访问系统设备，允许它们在实际方式和保护方式下运行。

语法：BIND infile 〖implibs〗 〖linklibs〗 〖/O outfile〗 〖/N@file〗 〖/N functions〗〖/M mapfile〗

在以上语法中，infile是OS/2应用程序的名字。implibs是一个或多个输入库名，linklibs是一个或多个标准库文件或目标文件名。

任选项：

/HELP	为BIND提供一个联机帮助。首先BIND试图执行Quick Help程序QH.EXE。如果Quick Help或它的数据库不存在，则/HELP选择项在标准输出设备上列出BIND语法和任选项。
/M 〖AP〗 mapfile	为可执行文件的实方式环境建立一个链接映象到指定的mapfile中。
/N 〖AMES〗 functions	允许列出仅在保护方式下支持的函数表，与函数表或由@开头定义的文件一起使用。
/NOLOGO	删除开始标题显示。
/O 〖UTFILE〗 outfile	为约束应用定义名称，outfile。
/?	显示BIND语法。

CL（编译器）

简述：CL 用于编译和链接一个或多个 C 源文件。

语法：CL〖option〗 〖filename〗... 〖libraries link-options〗

任选项

/A{T|S|M|C|L|H} 从这些标准存贮模式中选择一个：

任选项	说明
/AT	微小存贮模式，编码和数据限制在64K内，必须与CRTCOM>LIB 链接。为实方式生成一个.COM 文件(与/Asnd 一样)。
/AS	小存贮模式。编码和数据分别限制在64K内。(与/Asnd 一样)
/AM	中存贮模式。数据限制在64K内。(与/Alnd一样)
/AC	压缩存贮模式，编码限制在64K内。(与/Asfd一样)
/AL	大存贮模式。编码和数据没有限制。(与/Alfd一样)
/AH	超大存贮模式，与大存贮模式一样，但单个数据组可以超过 64K。(与/Alhd 一样)

/Astring 建立一个指定的存贮模式。string由3个任意排列的字符组成，指明编码、数据指针大小和段。

分组	代码	描述
代码指针	s	小模式 (Near)
	l	大模式 (Far)
数据指针	n	Near
	f	Far
	h	Huge
段	d	SS==DS
	u	SS==DS; 每进入一个函数装入DS。
	w	SS!=DS; 进入函数时不装入DS。

/B1〖path〗 生成一个叫做C1L.EXE可选择的预处理程序。使用这一选择项

	编译程序时，产生下列信息 the compiler is out of rear heap。C1L.EXE所在的驱动器和路径可由选择的 path 指定。
/B2〖path〗	生成一个两遍的可选择的编译程序C2L.EXE。
/B3〖path〗	生成一个三遍的可选择的编译程序C3L.EXE。
/C	当预处理一个文件时，保留注释内容；仅和/E，/P或?EP一起使用。
/c	不带链接编译。建立一个目标文件，但不可执行。
/D id〖=〖value〗〗	在预处理程序中定义一个含有id的符号。如果指定value，id的值就是 value。如果不等号给出没有 value 值，id 的值是空。如果只有 id 没有等号。id 的值是 1。
/E	预处理源文件，将结果拷贝到标准输出设备，并插入#line命令。
/EP	预处理源文件，将结果拷贝到标准输出设备，但不插入#line命令。
/F haxnum	建立栈的大小为hexnum个字节。(与/link/STACK: number一样)。数值必须用十六进制表示。
/Fa〖listfile〗	生成一个汇编表，如果没有指定listfile，/Fa缺省项是源文件名加上.ASM。不能与/qc 任选项一起使用。
/Fb bovmd-exe	生成一个约束的可执行文件。仅与/Lp一起使用。
/Fc 〖listfile〗	生成一个混合的源汇编代码表。如果没有指定listfile，/Fc认为是源文件名加上.COD，不能与/qc 一起使用。
/Fe exefile	可执行文件的名字。
/Fl 〖listfile〗	产生一个目标代码表。如果没有给出listfile。/Fl认为是源文件名加上.COD。不能与/qc 一起使用。
/Fm 〖mapfile〗	产生一个链接程序的映象文件。如果没有指定mapfile，/Fm认为是第一次使用的源文件名加上.MAP。
/Fo objfile	目标文件名。
/FPa	产生一个浮点调用，并选择一个可替换的数学库程序。不能与/qc 一起使用。
/FPc	产生一个浮点调用，并选择一个仿真程序库（如果有的话，使用 80x87 协处理器)。不能与/qc 一起使用。
/FPc87	产生一个浮点调用，并选择80x87程序库（运行时，需要一个80x87 协处理器)。不能和/qc 一起使用。
/FPi	生成80x87指令，并选择一个仿真程序库（如果有的话，使用80x87 协处理器)。这是/FP 任选项的省略形式。
/FPi87	生成80x87指令，并选择80x87程序库（运行时，需要一个80x87 协处理器)。

/Fr〖browsefile〗	生成一个标准的PWB Source Browser数据库。如果没有指定browsefile。/Fr认为是库名加上.SBR。
/FR〖browsefile〗	生成一个扩展的Source Browser数据库。如果没有指定browsefile，/FR认为是库名加上.SBR。
/Fs〖listfile〗	产生一个源列表。如果没有指定listfile，/Fs认为是源文件名加上.LST。不能与/qc一起使用。
/Fx〖xreffile〗	为Microsoft Macro Assembler (MASM)交叉引用文件指定文件名。如果没有定义xreffile，/Fx认为是源文件名加上.CRF。
/Go	生成8086/8088指令。这是/G任选项的缺省形式。
/G1	生成80186/80188指令。
/G2	生成80286指令。
/Gc	定义使用FORTRAN或Pascal类型函数调用和命名规范。
/Gd	定义标准（缺省项）C调用规范。
/Ge	允许调用栈校验程序（缺省项）。
/Gi	编译增量部份。（当和/qc一起使用时），只对已改动过的函数重新编译。没有/qc，/Gi增量部份由装填目标文件链接。隐含/Li。
/Gm	将字符串存在常数段中（CONST），不能与/qc一起使用。
/Gr	使新的fastcall函数能够调用正规函数。当可能时，数值被贮存在寄存器而不是栈中。
/Gs	压缩对栈校验程序的调用。
/Gf〖number〗	将大于或等于number字节的数据项放在一个新的段中，如果没有指定number，缺省值是256。
/Gw	生成适合在Microsoft Windows TM应用中使用的entry/exit代码系列。
/GW	类似于/Gw，但是生成更有效的entry系列，用于代码而不是用户的callback函数。
/H number	限定外部名为number个有意义的字符，缺省值是31个字符，不能和/qc一起使用。
/HELP	调用Quick Help实用程序，如果没有Quick Help程序，CL在标准输出设备上显示最常用的任选项。
/I directory	将directory加入到目录表的前面，以便能够找到所包括的文件。
/J	将带符号的缺省值改为不带符号的char类型。
/Lc	使链接程序建立一个兼容式的可执行文件。类似于/Lr。
/Li〖number〗	产生一个增量的链接程序[LINK]，而不是一个标准的链接程序LINK。在建立一个更大的可执行文件时，ILINK比LINK运行要快。number定义了字节数，在这些字节里，链接程序装

填所有相邻近的函数。

/Lp　使链接程序生成一个保护式的可执行文件。

/Lr　使链接程序生成一个real-mode可执行文件。

/link link-libinfo　将链接程序的任选项或在link-libinfo中的库文件名传送给LINK。

/MAMASM option　将指定的任选项传送到Microsoft Macro Assembler (MASM)。为带有扩展名.ASM的命令行中列出的文件自动生成MASM。

/MD　建立一个动态链接C运行库（只适用于OS/2），相当于/ALw /FPi /G2 /DDLL /DMT。没有库搜索记录。

/ML　作为动态链接库的一部份，静态地链接C运行库程序（只适用于OS/2），相当于/ALw/FPa/G2/DMT。库搜索记录转换到LLIBCDLL.LIB。

/MT　允许支持多流处理程序（仅适合于OS/2），相当于/ALw /FPi /G2/DMT库搜索记录转换到LLIBCMT.LIB。

/NDdataseg　设置数据段名。

/NMmodule　设置模块名。

/nologo　禁止开始指令标题的显示。

/NTtextseg　设置编码段名。

/O〖opt codes〗　控制优化，如果没有代码，使用缺省类型优化。任选项opt codes可以有一个或多个下列字符:

代码	描述
a	假设没有别名使用。
c	允许缺省本地公用表达式，(块一级)。
d	禁止所有的优化。
e	允许全局寄存器分配。
g	允许全部优化和全部公用表达式。
i	允许内部程序产生。
l	允许循环优化。
n	禁止不安全的循环优化（缺省值)。
p	改进浮点运算的一致性。
r	禁止从函数中的内部返回。
s	支持更小的代码长度。
t	支持更快的执行速度（缺省值)。
w	除非函数调用，否则没有别名使用。(不能与/qc一起使用)。

x	最大优化。(相当于／Oecilgt ／Gs)。
z	允许最大循环和全局量寄存器分配优化。
／P	预处理源文件，并将输出送到源文件名后加一个扩展名。I 的文件中。(basename.I)。
／qc	产生一个快速编译。在快速编译中，下列的任选项会产生错误：／Fa，／Fc，／Fl，／FPa，／FPc，／FPc87，／Fs，／Cm，／H，／Ow，／Zc。
／Sl linewidth	设置源程序表每行的字符宽度范围为79–132，缺省值为 79。
／Sp pagelength	设置源程序表在每一页里有多少行。范围为15–255，省缺值为 63。
／Ss Subfifle	为源程序表定义Subfifle
／St fifle	为源程序表定义title。
／Ta asm–srcfile	将asm–srcfile作为一个汇编程序源文件来处理，而不管是否有.ASM 扩展名。
／Tc c–srcfile	指出c–srcfile是C源文件。不管是否有.C扩展名。
／u	删除所有已定义的标识符的含意。
／U identfier	删除已定义并已给出的标识符的含意。
／V string	将版本string拷贝到目标文件中。
／w	压缩编译程序报警信息； 相当于／WO。
／W{0|1|2|3|4}	为编译程序报警信息设置输出等级。缺省值是1。
／WX	使所有的报警为致命错误，当报警发生时，不产生目标文件。
／X	在嵌入文件中查找，勿略掉“标准位置”上的表。
／Za	使用ANSI语言，禁止Microsoft C的任何扩展。
／Zc	使说明为 Pascal的函数不做特别处理。不能与／qc 一起使用。
／Zd	为SYMDEB调试程序生成行数信息。
／Ze	允许对Microsoft C扩展定义。这是／E任选项的缺省形式。
／Zg	从函数定义中生成函数原型，并且将说明输出到标准输出设备上。不编译程序。
／Zi	生成Microsoft CodeView面向窗口调试程序所需要的符号信息。
／Zl	在目标文件中压缩库查找记录。

/Zp〖{1|2|4}〗	压缩已定义字节边界上的结构成员。
/Zr	为检查空指针和溢出指针生成代码。(在CL命令中，只和/qc一起使用)。
/Zs Sourcefiles	仅进行一个语法检查。

CodeView

简述: Micorsoft CodeView]窗口调试程序运行一个已编译的程序，同时显示它的源代码，程序变量，存贮区的位置，处理机寄存器和其它的相关信息。

语法: CV 〖options〗]executablefile 〖arguments〗。

为调试一个保护式程序，在你的CONFIG.SYS中要建立IOPL＝YES，并使用下列的语法:

CVP 〖options〗 executablefile 〖arguments〗

任选项:

/2	允许使用两个监督程序。
/25	在25行方式下启动。
/43	在EGA43行方式下启动。
/50	在VGA 50行方式下启动。
/B	在CGA或EGA黑白方式下启动。
/C commands	在启动后执行commands。
/D 〖buffersize〗	允许磁盘复盖（仅适用于DOS)。
/E	允许Expanded Memory Support(EMS)（仅适用于DOS)。
/F	在CodeView和你调试的程序之间不交换视频页面。通过两个视频页面来交换调试信息和屏幕输出。(比/S快)。
/I number	转换不可屏蔽的中断。打开8259中断陷井（/I1）或关闭(/I0)。
/K	禁止为正在调试的程序装入键盘监督程序。
/L dynlib	允许CodeView查找OS/2动态链接库程序的符号信息。
/M	禁止CodeView对鼠标的支持（当调试一个支持鼠标的应用程序时，使用这个选择项)。
/N number	/NO 使CodeView陷入； /N1与此相反。
/O	在OS/2保护方式下允许多处理方式调试。
/R	允许使用80386调试寄存器（在OS/2下不允许)。
/S	开始屏幕交换。(通过改变缓冲区来交换屏幕，与图形程序一起使用)。
/X	使用扩展存贮区来增加调试功能（仅适合于DOS)。

CodeView 命令

操作	键操作	鼠标操作
显示相应的帮助信息	F1	选help菜单
显示帮助内容	SHIFT+F1	选Help菜单上的Contents命令
转到下一个帮助屏幕	CTRL+F1	——
转到上一个帮助标题	SHIFT+CTRL+F1	——
转到前一个帮助屏幕上	ALT+F1	在帮助屏幕中选Back
变换寄存器窗口	F2	选View菜单中咸Regisfers
切换源程序／汇编程序／混合方式	F3	选Options菜单上的Source Window命令。
变换存贮区窗口格式	SHIFT+F3	选Options菜单Memory Window命令。
转到输出屏幕	F4	选View菜单Output命令。
关闭窗口	CTRL+F4	在窗口左上角处按键
转到下一个断点或程序末尾	F5	在状态行的GO处LEFT
转到下一个窗口	F6	选所希望的窗口
转到先前的窗口	SHIFT+F6	选所希望的窗口
执行到光标处	F7	在状态行上按Right键相应位置上
追踪进入子过程	F8	在Trace上按Left键
显示记录下的先前命令	SHIFT+F8	——
改变窗口大小	CTRL+F8	在窗口的连界和要变动的地方按Left 键
转到光标所在行的断点	在相应位置上用F9	在相应的位置上和要改变的地方按两下 Left 键
越过子过程	F10	在Step上面按Left键
显示记录的下一个命令	SHIFT+F10	——
增大窗口	CTRL+F10	在窗口右上角处按键
改变寄存器窗口中的标记	任何打印字符	在标记上按两下Left键
删除光标处的字符	DEL	——
插入和重敲模式转换	INS	——
将文本拷贝到删除缓冲区	CTRL+INS	在Edit菜单的Copy命令上按键
从删除缓冲区中剪贴文本	SHIFT+INS	在Edit菜单的Paste命令上按键
移动到下一条命令（仅对Command 窗口）	TAB	在相应的地方按Left键

移到上一命令（仅对Command窗口）	SHIFT+TAB 在相应的地方按LEFT键
寻找选择的文本	CTRL+＼在Search菜单的Selected Text命令上按键。
重复上一次查找	ALT+／在Search菜单的Repeat Find命令上按键
加入Watch表达式	CTRL+W 在Watch菜单的Add Watch命令上按键
删除Watch表达式	CTRL+U 在Watch菜单的Delete Watch命令上按键
为变量打开Quick Watch窗口	SHIFT+F9 在Watch窗口的Quick Watch命令上按键
在窗口中向上滚动一行	CTRL+UP 在滚行的向上箭头处按一下Left按钮
在窗口中向下滚动一行	CTRL+DOWN 在滚行的向下箭头处按Left按钮
在窗口中向上滚动一页	PGUP 在垂直杆的上部按Left键
在窗口中向下滚动一页	PGDN在垂直杆的下部按Left键
将窗口向左滚动	CTRL+PGUP在左箭头处或水平杆的左边沿按Left键
将窗口向右滚动	CTRL+PGDN在右箭头处或水平杆的右边沿按Left键
移动光标到行的开始	HOME在相应位置上按Left键
移动光标到行的末尾	END拉垂直杆到底部
滚到文件的顶部	CTRL+HOME拉垂直杆到顶部
滚到文件的末尾	CTRL+END拉垂直杆到底部
将光标移一个字	CTRL+LEFT在相应的位置上按Left键 或CTRL+RIGHT
将光标移一行	UP或DOWN在相应的位置上按Left键