

Microsoft C 6.0

高级程序设计技术

北京科海培训中心

Microsoft C 6.0 之二

高级程序设计技术

北京科海培训中心

发行：北京科海培训中心资料组

地址：北京海淀路 82 号科海培
训中心资料组

乘车：332、320 黄庄站下车海淀
文化馆平房

电话：2562449 2562954

介 绍

《高级编程技巧》一书描述了利用 Microsoft C 高级开发系统新提供的集成环境——Microsoft 程序员工作台和源语言级调试工具 CodeView 调试器开发程序的技巧。

本手册介绍了如何组合 Microsoft C 高级开发系统中的各种工具和实用程序以获得一个最强大的开发环境。用好高级开发系统的关键是要对它进行合理得剪裁，使其适合于你的需要。

此书按专题的形式组织，针对使用 Microsoft C 6.0 时出现的一些问题进行讨论，而没有罗列出各个选择项。有关高级开发系统中 CodeView 调试器、程序员工作台所涉及的菜单项及一些实用程序命令，请参阅 Microsoft C Advisor（联机帮助）或 C 参考手册。

本书的范围

《高级编程技巧》分为四个部分，第一部分，“提高程序的性能”能帮助你写出更高效的程序，它详细讲解了优化问题。· 即何时和为什么要使用各种优化选择项。这部分还介绍了新的存储管理选择项及在何时使用它们。如在第三章中，讲解了嵌入式汇编语言。这是一个新的特性，它允许在 C 语言中使用汇编语言编程。

第二部分，“提高程序的效率”，将帮助你使程序运行的更快，效率更高。第 8 章介绍裁剪程序员工作台的各种不同的方法。程序员工作台（PWB）是一个包括编辑器的集成开发环境，使用它可以：

- .建立新程序
- .修改已存在的程序
- .Browse 源程序
- .获得 PWB、C 语言及 C 运行时间库的帮助信息
- .建立程序的 Build 列表
- .重新编译（Build）程序
- .使用 CodeView 调试器调试程序

第 8 章还介绍了如何通过改变键盘的值、定义宏及定义 C 扩展来改变 PWB 的性能以适合于你的编程风格。

第二部分还讲解了用于程序维护的新的实用程序 · NMAKE。NMAKE 允许你输入程序名列表，这使得在重新编译程序时有很大的灵活性。NMAKE 是 Microsoft XENIX MAKE 的超级子集，因而它比过去版本中的 MAKE 功能更强。

第二部分中的第 9 章描述了 CodeView 调试器，CodeView 3.0 提供了许多新的功能，包括记录一个调试过程然后重新调试（与过去有关的、动态的调试）。

第三部分，“特设环境”描述了新的图形功能。此外，还介绍了如何用多种语言混合编程，提供了 tips 使你的程序更简明。使用 Microsoft C 能很容易地建立图形应用程序。

Microsoft C 运行时间库包括低级图形操作功能，如：画线、矩形及圆，还包括一些建立图示的功能，如：饼图及直方图。

第四部分，“OS/2 支持”介绍了如何使用高级开发系统建立 OS/2 应用程序。第四部分中有三章分别介绍了双向应用程序、multithread 应用程序及动态链接库的建立。

目 录

介绍	(1)
----------	-----

第一部分 提高程序的性能

第一章 优化 C 程序	(3)
1.1 从程序员工作台控制优化	(3)
1.2 从命令行控制优化	(3)
1.3 用编译杂注控制优化	(4)
1.4 缺省的优化	(5)
1.4.1 消除公共子表达式	(5)
1.4.2 消除死存储	(5)
1.4.3 常数传送	(5)
1.5 剪裁优化	(6)
1.5.1 优化速度或空间 (/Ot 和 /Os)	(6)
1.5.2 产生内部函数 (/Oi)	(6)
1.5.3 假设无别名使用 (/Oa 和 /Ow)	(9)
1.5.4 执行循环优化 (/Ol)	(11)
1.5.5 禁止不安全的循环优化 (/Oz)	(12)
1.5.6 允许深度优化 (/Oz)	(12)
1.5.7 取消栈探查 (/Gs)	(13)
1.5.8 允许分配全局寄存器 (/Oe)	(13)
1.5.9 允许公共子表达式优化 (/Oc 和 /Og)	(14)
1.5.10 实现浮点结果的一致性 (/Op)	(14)
1.5.11 使用 80186,80188 及 80286 处理器 (/G0、/G1、/G2)	(15)
1.5.12 最大效率的优化 (/Ox)	(15)
1.6 控制优化的链接程序 (LINK) 选择项	(16)
1.6.1 允许远调用优化 (/FARCALL TRANSLATION)	(16)
1.6.2 压缩代码 (/PACKCODE)	(17)
1.6.3 压缩数据 (/PACKDATA)	(17)
1.6.4 压缩可执行文件 (/EXEPACK)	(18)
1.7 不同环境中的优化	(18)
1.7.1 DOS 中的优化	(18)
1.7.2 OS/2 中的优化	(18)

1.7.3 Microsoft Windows 中的优化	(18)
1.8 选择函数的调用形式	(18)
1.8.1 C 调用形式	(18)
1.8.2 FORTRAN / Pascal 调用形式	(19)
1.8.3 寄存器调用形式	(19)
1.8.4 __fastcall 调用形式	(19)
第二章 内存管理	(22)
2.1 指针的大小	(22)
2.1.1 指针及 64K 的段	(22)
2.1.2 近 (near) 指针	(22)
2.1.3 远 (far) 指针	(23)
2.1.4 巨型 (huge) 指针	(23)
2.1.5 基址导址	(24)
2.2 选择标准存储模式	(24)
2.2.1 六种标准存储模式	(24)
2.2.2 对代码和数据大小的限制	(25)
2.2.3 微型存储模式	(25)
2.2.4 巨型存储模式	(25)
2.2.5 空指针	(26)
2.2.6 存贮模式的说明	(27)
2.3 混合存储模式	(27)
2.3.1 有关指针的问题	(29)
2.3.2 说明近、远、巨型及其变量	(29)
2.3.3 说明近函数和远函数	(30)
2.3.4 指针转换	(31)
2.4 自定义存储模式	(33)
2.4.1 设置代码指针的长度	(33)
2.4.2 设置数据指针的长度	(34)
2.4.3 设置段	(34)
2.4.4 支持自定义模式的库	(36)
2.4.5 设置数据词	(37)
2.4.6 命名模块和段	(37)
2.4.7 说明正文和数据段	(38)
2.5 基本变量的使用	(38)
2.5.1 新引进的几个关键字	(38)
2.5.2 说明基变量	(39)
2.5.3 基指针的优点	(44)

第三章 使用嵌入式汇编语言程序	(46)
3.1 嵌入式汇编语言的优点	(46)
3.2 __asm 关键字	(46)
3.3 __asm 中的汇编语言	(47)
3.4 在__asm 块中使用 C	(49)
3.4.1 使用操作符	(49)
3.4.2 使用 C 的符号	(50)
3.4.3 访问 C 数据	(50)
3.4.4 编写函数	(51)
3.5 使用及保存寄存器	(52)
3.6 跳转到标号处	(53)
3.7 调用 C 函数	(54)
3.8 将__asm 块定义为宏	(55)
3.9 优化	(56)
第四章 控制浮点数学运算	(58)
4.1 说明浮点类型	(58)
4.1.1 用浮点类型说明变量	(58)
4.1.2 说明函数的返回值为浮点类型	(59)
4.2 C 运行时间库对 long double 类型的支持	(60)
4.3 数学包概述	(60)
4.3.1 仿真包	(60)
4.3.2 数学协处理器包	(61)
4.3.3 补充数学包	(61)
4.4 选择浮点选择项 (/FP)	(61)
4.4.1 嵌入式仿真包选择项 (/FPi)	(63)
4.4.2 嵌入式数学协处理器指令选择项 (/FPi87)	(63)
4.4.3 调用仿真包选择项 (/FPc)	(63)
4.4.4 调用数学协处理器选择项 (FPc87)	(64)
4.4.5 使用补充数学库选择项 (/FPa)	(64)
4.5 浮点选择项涉及的库	(65)
4.5.1 使用标准库链接	(65)
4.5.2 嵌入指令及调用所涉及的库	(65)
4.6 浮点选择项的兼容问题	(65)
4.7 使用 NO87 环境变量	(66)
4.8 不兼容问题	(66)

第二部分 提高程序员的工作效率

第五章 快速编译及链接	(71)
5.1 快速编译	(71)
5.1.1 快速编译程序	(71)
5.1.2 可增编译选择项	(71)
5.2 使用 ILINK 快速链接	(71)
5.2.1 使用可增链接的准备工作	(72)
5.2.2 可增性错误	(73)
第六章 使用 NMAKE 实用程序开发项目 (project)	(74)
6.1 NMAKE 概况	(74)
6.2 NMAKE 命令	(74)
6.3 NMAKE 描述文件	(75)
6.3.1 描述块	(75)
6.3.2 注释	(79)
6.3.3 宏	(79)
6.3.4 推理规则	(85)
6.3.5 指令	(87)
6.3.6 伪目标	(89)
6.3.7 程序员工作台的 extmake 语法	(90)
6.4 命令行选择项	(91)
6.5 NMAKE 命令文件	(92)
6.6 TOOL.INI 文件	(92)
6.7 嵌入文件	(93)
6.8 NMAKE 的操作顺序	(94)
6.9 NMAKE 与 MAKE 的不同之处	(95)
第七章 用 HELPMAKE 建立 Help 文件	(97)
7.1 Help 数据的结构和内容	(97)
7.1.1 Help 文件的内容	(97)
7.1.2 Help 文件的格式	(98)
7.2 调用 HELPMAKE	(99)
7.3 HELPMAKE 选择项	(99)
7.3.1 编码选择项	(100)
7.3.2 解码选择项	(102)
7.4 建立 Help 数据库	(103)
7.5 Help 正文规范	(103)
7.5.1 Help 正文件结构	(103)

7.5.2 局部局境	(104)
7.5.3 局境前缀	(104)
7.5.4 超链式	(104)
7.6 使用 Help 数据库格式	(107)
7.6.1 quickHelp 格式	(107)
7.6.2 最小格式化的 ASCII 格式	(111)
7.6.3 多种的正文格式 (RTE)	(112)
第八章 裁剪 Microsoft 程序员工作台	(114)
8.1 设置开关	(114)
8.1.1 编辑 <assign> 伪文件	(114)
8.1.2 编辑 TOOLS.INI 初始化文件	(115)
8.2 键指派	(115)
8.3 编写宏	(116)
8.3.1 宏语法	(116)
8.3.2 宏响应	(117)
8.3.3 宏参数	(118)
8.3.4 宏条件	(118)
8.3.5 临时宏	(119)
8.3.6 记录宏	(119)
8.4 编写并建立 C 扩充	(120)
8.4.1 建立实方式的 C 扩充	(122)
8.4.2 建立保护方式的 C 扩充	(123)
8.4.3 描述功能和开关	(124)
8.4.4 初始化功能	(125)
8.4.5 原型功能	(126)
8.4.6 接收参数	(126)
8.4.7 调用 PWB	(128)
8.4.8 调用 C 库函数	(130)
第九章 用 CodeView 调试 C 程序	(132)
9.1 CodeView 窗口	(132)
9.2 调试技术概述	(133)
9.3 查看并修改程序数据	(134)
9.3.1 在 Watch 窗口显示变量	(134)
9.3.2 在 Watch 窗口显示表达式	(134)
9.3.3 显示数组和结构	(135)
9.3.4 动态地显示数组元素	(136)
9.3.5 使用 Quick Watch	(137)
9.3.6 显示内存	(137)

9.3.7 显示寄存器	(138)
9.3.8 修改变量、寄存器及内存的值	(138)
9.4 控制执行	(139)
9.4.1 连续执行	(139)
9.4.2 单步执行	(141)
9.5 重新显示一个调试阶段	(141)
9.6 高级 CodeView 技术	(142)
9.7 用命令行选择控制 CodeView	(144)
9.8 用 TOOLS.INI 文件裁剪 CodeView	(145)

第三部份 特殊环境

第十章 编写图形程序	(149)
10.1 显示方式	(149)
10.1.1 简单的图形程序	(149)
10.1.2 设置显示方式	(150)
10.1.3 读取显示方式信息	(151)
10.1.4 最高分辨率与最大颜色数	(152)
10.1.5 选择显示方式	(153)
10.2 调制颜色与调色板的改变	(153)
10.2.1 CGA 调色板	(154)
10.2.2 Olivetti 调色板	(155)
10.2.3 VGA 调色板	(155)
10.2.4 MCGA 调色板	(156)
10.2.5 EGA 调色板	(156)
10.2.6 符号常量	(157)
10.3 在坐标系中指定一点	(157)
10.3.1 物理坐标系	(158)
10.3.2 视见区坐标系	(159)
10.3.3 窗口坐标系	(160)
10.3.4 屏幕位置	(161)
10.3.5 外接矩形	(161)
10.3.6 象素光标	(162)
10.4 图形函数	(162)
10.4.1 控制显示方式	(162)
10.4.2 改变颜色	(163)
10.4.3 画点, 线和图形	(164)
10.4.4 定义模式	(165)
10.4.5 对图象进行操作	(165)

10.5 使用图形字型	(166)
10.5.1 使用 C 字型库	(167)
10.5.2 登记字型	(167)
10.5.3 设置当前字型	(167)
10.5.4 显示文本	(169)
10.5.5 例子程序	(169)
10.5.6 有效地使用字型	(171)
第十一章 产生图表及图形.....	(172)
11.1 表意制图综述	(172)
11.2 有关图表的说明	(173)
11.3 编写一个表意图形程序	(175)
11.3.1 Pie 图表	(175)
11.3.2 Bar, Column 和 line 图表	(177)
11.3.3 Scatter 图	(181)
11.4 控制颜色和模式	(183)
11.4.1 颜色池	(184)
11.4.2 格式池	(184)
11.4.3 模式池	(185)
11.4.4 字符池	(186)
11.5 设置图表环境	(186)
11.5.1 titletype 结构	(187)
11.5.2 axistype 结构	(188)
11.5.3 windowtype 结构	(190)
11.5.4 Legendtype 结构	(191)
11.5.5 chartenv 结构.....	(192)
第十二章 混合语言程序设计.....	(194)
12.1 混合语言调用	(194)
12.2 语言约定要求	(195)
12.2.1 命名约定要求	(195)
12.2.2 调用约定要求	(197)
12.2.3 参数传递要求	(199)
12.3 编译和链接	(199)
12.3.1 用正确的存储模式编译	(200)
12.3.2 与语言库链接	(200)
12.4 C 对高级语言的调用	(200)
12.5 C 调用 BASIC	(202)
12.6 C 调用 FORTRAN	(205)
12.6.1 从 C 调用 FORTRAN 子例程	(205)

12.6.2 从 C 调用 FORTRAN 函数	(206)
12.7 C 调用 PASCAL	(207)
12.7.1 从 C 调用一个 PASCAL 过程	(207)
12.7.2 从 C 调用 PASCAL 函数	(208)
12.8 C 调用汇编	(209)
12.8.1 编写汇编语言过程	(210)
12.8.2 建立过程	(210)
12.8.3 进入过程	(211)
12.8.4 为局部数据分配空间	(211)
12.8.5 保留寄存器值	(212)
12.8.6 访问参数	(212)
12.8.7 返回一个值	(214)
12.8.8 退出过程	(215)
12.9 混合语言程序设计中的数据处理	(216)
12.9.1 缺省的命名和调用约定	(216)
12.9.2 数值数据表示	(217)
12.9.3 串	(217)
12.9.4 数组	(220)
12.9.5 数组说明和加下标	(220)
12.9.6 结构、记录和用户定义类型	(221)
12.9.7 外部数据	(222)
12.9.8 指针和地址变量	(222)
12.9.9 公用块	(223)
12.9.10 使用可变数目的参数	(224)
第十三章 编写可移植的程序	(225)
13.1 硬件约定 (Assumption)	(225)
13.1.1 基本数据类型的大小	(225)
13.1.2 内存分配的顺序与字节对准	(227)
13.1.3 字中的字节顺序	(230)
13.1.4 结构的读和写	(231)
13.1.5 结构中的位域	(231)
13.1.6 处理器运算的模式	(233)
13.1.7 指针	(233)
13.1.8 地址空间	(235)
13.1.9 字符集	(235)
13.2 编译程序的约定	(236)
13.2.1 符号扩展	(236)
13.2.2 标识符的长度与大小写	(238)

13.2.3 寄存器变量	(239)
13.2.4 参数个数可变的函数	(239)
13.2.5 求值顺序	(240)
13.2.6 函数与宏参数的副作用	(241)
13.2.7 环境差异	(241)
13.3 数据文件的可移植性	(241)
13.4 考虑 Microsoft C 特性的可移植性	(241)
13.5 Microsoft C 字节顺序	(241)

第四部份 OS/2 支持

第十四章 建立 OS/2 的应用程序	(245)
14.1 OS/2 应用程序接口 (API)	(245)
14.1.1 调用 OS/2 API	(245)
14.1.2 包含 OS/2 头文件	(246)
14.1.3 建立双模式程序作为簇应用 (Family Applications)	(247)
14.2 CL 命令的编译选择项	(249)
14.2.1 链接方式选择项 (/Lp, /Lr, 和 /Lc)	(249)
14.2.2 建立联编程序的选择项 (/Fb)	(249)
14.2.3 库选取选择项 (/MT, /ML, /MD, /ZI)	(250)
14.2.4 内存模式选择项 (/AX)	(251)
14.3 模块定义文件和输入库	(252)
14.3.1 在 LINK 命令中加入模块定义文件	(253)
14.3.2 建立动态链接库 (Dlls)	(253)
14.3.3 建立具有 I/O 特权的程序	(253)
14.3.4 建立表示管理程序的应用程序	(254)
14.3.5 用 IMPLIB 实用程序建立输入 (import) 库	(255)
14.4 LINK 命令选择项	(255)
14.5 BIND 实用程序	(256)
第十五章 建立多线索的 OS/2 应用程序	(258)
15.1 多线索的程序	(258)
15.1.1 库支持	(258)
15.1.2 包含文件	(260)
15.1.3 为线索控制提供的 C 运行时间库函数	(260)
15.2 多线索的 C 程序例子	(261)
15.3 编写多线索的程序	(267)
15.4 编译和链接	(269)
15.5 常见的问题及可能的原因	(270)

15.6 使用保护方式的 CodeView 调试器	(271)
15.6.1 用 / Zi 选择项编译	(271)
15.6.2 线索号提示	(271)
15.6.3 线索命令	(271)
15.6.4 CodeView 使用的屏幕组	(273)
第十六章 OS/2 的动态链接	(274)
16.1 动态链接概念	(274)
16.1.1 装入时和运行时链接	(274)
16.1.2 应用程序和 DLLS	(275)
16.1.3 DLL 和 Microsoft C 运行时库	(275)
16.2 设计和编写 DLLS	(277)
16.2.1 浮点数学要求	(277)
16.2.2 初始化及终止要求	(278)
16.2.3 使 DLL 要重入	(280)
16.2.4 信号处理	(282)
16.2.5 使用 Microsoft C 关键字	(282)
16.2.6 DLL 的编译选择项	(283)
16.3 用 Microsoft C 建立动态链接 (DLL)	(284)
16.3.1 调用静态 C 运行时库函数的动态链接库 (DLL)	(284)
16.3.2 不调用 C 运行时库的 DLLS	(287)
16.3.3 调用 C 运行时动态链接库的应用程序和 DLL	(288)
16.3.4 使用 CodeView 调试动态链接库 (DLL)	(291)

附 录

附录 A 退出码的使用	(295)
A.1 退出函数	(295)
A.2 从命令和批文件中检测退出码	(296)
A.3 从其它程序访问退出码	(296)
附录 B C5.1 和 C6.0 版本之间的差异	(298)
B.1 为与 ANSI 标准一致所做的修改	(298)
B.1.1 ANSI-Mandated 新特征	(298)
B.1.2 整数改进规则	(298)
B.1.3 定义 NULL 为一指针	(299)
B.1.4 移位算子	(299)
B.1.5 指向类型定义的指针	(299)
B.1.6 标识非标准的关键字	(300)
B.1.7 三字母表示的字符	(300)

B.1.8	与 ANSI C 语言标准不一致之处	(300)
B.2	新关键字和函数	(300)
B.2.1	直接插入汇编程序	(301)
B.2.2	基指针和对象	(301)
B.2.3	基准分配支持	(302)
B.2.4	释放不使用的堆空间	(302)
B.2.5	将静态数据空间变为堆	(302)
B.2.6	长双精度类型	(302)
B.2.7	长双精度函数	(302)
B.2.8	不依赖于存储模型的串函数和内存函数	(303)
B.2.9	混合存储模式的内存分配支持	(303)
B.2.10	fastcall 属性 (/Gr 选项)	(303)
B.2.11	驱动器和目录函数	(304)
B.2.12	为使用 OS/2 而增加的文件输出函数	(304)
B.3	新的特征	(305)
B.3.1	串和宏	(305)
B.3.2	CL 选项	(305)
B.3.3	微模式 (.COM 文件)	(305)
B.3.4	优化编译提示	(306)
B.3.5	无名结构和联合	(306)
B.3.6	无长度的数组作为一个结构的最后成员	(307)
B.3.7	改进的警告信息	(307)
B.3.8	宏	(307)
B.3.9	改进了对 OS/2 中多个执行路径的支持	(307)
B.3.10	对 OS/2 中的管道支持	(308)
B.4	代码生成方面的差别	(308)
B.4.1	速度和空间的改进	(308)
B.4.2	代码质量	(308)
B.4.3	浮点代码生成	(308)
B.4.4	内部函数	(308)
B.5	变化和删除	(309)
B.5.1	被删除的特征	(309)
B.5.2	实表达式求值	(309)
B.5.3	缺省优化	(309)
B.5.4	char 型参数的符号扩展	(310)
B.5.5	条件编译	(310)
B.5.6	const 和 volatile 限定词	(310)
B.5.7	内存分配	(310)
B.5.8	被命令行参数使用的内存	(310)

B.5.9 printf 的格式说明符	(310)
B.5.10 返回浮点值的函数	(311)
附录 C 由编译实现定义的特性	(312)
C.1 翻译	(312)
C.1 翻译	(312)
C.1.1 诊断	(312)
C.2 环境	(312)
C.2.1 函数 main 的参数	(312)
C.2.2 交互设备	(313)
C.3 标识符	(313)
C.3.1 不与外部连接的标识符的有效字符的个数	(313)
C.3.2 与外部连接的标识符的有效字符的个数	(313)
C.3.3 大小写字符	(313)
C.4 字符集	(313)
C.4.1 ASCII 字符集	(313)
C.4.2 多字节字符	(313)
C.4.3 每个字符的位数	(314)
C.4.4 字符集	(314)
C.4.5	(314)
C.4.6 宽字符	(314)
C.4.7 变换多字节字符	(314)
C.4.8 字符值的范围	(314)
C.5 整数	(317)
C.5.1 整数值的范围	(317)
C.5.2 整数的降级	(317)
C.5.3 符号整数的位操作	(317)
C.5.4 余数	(317)
C.5.5 右移	(318)
C.6 浮点算术	(318)
C.6.1 值	(318)
C.6.2 将整数变为浮点数	(318)
C.6.3 浮点值的截取	(318)
C.7 数组和指针	(318)
C.7.1 最大的数组长度	(318)
C.7.2 强制指针转换	(318)
C.7.3 指针减法	(319)
C.8 寄存器	(319)
C.8.1 寄存器的可用性	(319)