

计算机 C/C++ 语言系列丛书



主编 李春葆

# BORLAND C++ 3.1

# 使用大全

希望

学苑出版社

计算机C/C++语言系列丛书

# BORLAND C++3.1

## 使用大全

主 编

李春葆

编 著

张植民 陈曼铃

唐上达 马玉琳

学 苑 出 版 社

1993

(京)新登字 151 号

### 内 容 提 要

本书共分三个部分,分别介绍了 Borland C++ 程序设计, Borland C++ 参考指南和 Borland C++ 程序设计工具,由浅入深地叙述了 Borland C++ 环境,面向对象程序设计的基本概念, Borland C++ 运行库函数的详细参考信息以及 Turbo Assembler、Turbo Debugger 和 Turbo Profiler 等程序设计工具的有关功能及特征。书中实例丰富,层次分明,各部分相对独立,是程序员进行程序设计及系统开发的重要参考资料。

欲购本书的用户,请直接与北京 8721 信箱联系 邮编:100080,电话:2562329

计算机 C/C++ 语言系列丛书

**BORLAND C++ 3.1 使用大全**

---

主 编:李春葆

编 著:张植民 陈曼铃

唐上达 马玉琳

责任编辑:徐建军

出版发行:学苑出版社 邮政编码:100032

社 址:北京市西城区成方街 33 号

印 刷:双青印刷厂印刷

开 本:787×1092 1/6

印 张:55.625 字数:1289 千字

印 数:1—5000 册

版 次:1993 年 10 月第 1 版第 1 次

ISBN7-5077-0875-6/TP·24

本册定价:68.00 元

---

学苑版图书印、装错误可随时退换

# 前 言

本书的目的旨在提供 Borland C++ 运行库函数和 Borland C++ 的 Turbo Assembler、Turbo Profiler 及 Turbo Debugger 等工具的参考信息,以便于程序员的程序设计及系统开发。全书分为三大部分。第 I 部分为 Borland C++ 程序设计,包括前四章,介绍了 Borland C++ 环境,提供了 C、C++ 和面向对象程序设计的有关信息。第 II 部分为 Borland C++ 参考指南,包括第五章到第十九章,全面讨论了各种运行库函数的有关信息,第十九章提供了这些函数的应用例子。第 III 部分为 Borland C++ 程序设计工具,包括第二十章到第二十二章,介绍了用于提高 C++ 编译器功能的程序设计工具。

各章的简要描述如下:

## • Borland C++ 程序设计

第一章介绍了 Borland C++ 环境,提供了菜单和命令行任选项信息;第二章描述了 C 语言的优点、不足之处和代码结构;第三章讨论了有关面向对象程序设计的概念,包括封装,继承和多态性;第四章讨论了 C++ 语言及其在 C 上的扩充。

## • Borland C++ 参考指南

从第五章开始是参考指南部分,本章全面提供了 Borland C++ 运行库函数中字符分类宏方面的参考信息;第六章介绍了有关数据转换函数;第七章充分描述了 Borland C++ 运行库中的各个目录控制函数;第八章简述了图形程序设计的原理和有关函数的参考信息;第九章提供了输入/输出函数的有关信息;第十章描述了 DOS、BIOS 或 8086 的接口函数;第十一章讨论了内存和字符串操作函数;第十二章讨论了数学函数;第十三章描述了动态内存分配的基本原理和 Borland C++ 的动态内存分配函数;第十四章讨论了进程控制函数;第十五章介绍显示和操作文本窗口的有关函数;第十六章描述了 C++ 标准函数;第十七章讨论了时间和日期函数;第十八章讨论了所有不能归纳到前面第五章到第十七章中的其它函数;第十九章列举了所有 Borland C++ 库函数的应用实例。

## • Borland C++ 程序设计工具

本书的最后一部分起始于第二十章,这部分介绍了 Turbo Assembler、Turbo Debugger 和 Turbo Profiler 的有关功能。第二十章讨论了 Turbo Assembler 的命令行任选项、处理器指令、协处理器指令、目录、运算符和预定义符号等;第二十一章介绍 Turbo Debugger 的命令任选项和菜单信息;第二十二章描述了 Turbo Profiler,它是一个帮助我们优化程序性能分析的软件工具,在该章中详细地列出了命令行任选项,并解释了其环境菜单和各菜单任选项的功能,以便于掌握 Turbo Profiler 的很多特征。

由于本书篇幅较大,时间短促,错误之处难以避免,请广大读者不吝指正。

本书的出版得到北京希望电脑公司和武汉测绘科技大学计算机系的大力支持,特别是秦人华和宋明华两位老师提出过有建设性的意见,编者在此一并表示衷心的感谢。

编 者

1993 年 10 月于武汉

# 目 录

## 第一部分 Borland C++程序设计

第一章 Borland C++引言 .....	1
1.1 命令行编译器 .....	1
1.2 IDE .....	7
1.3 ≡或系统菜单 .....	8
1.4 File 菜单 .....	9
1.5 Edit 菜单.....	10
1.6 Search 菜单.....	10
1.7 Run 菜单.....	12
1.8 Compiler 菜单.....	13
1.9 Debug 菜单.....	14
1.10 Project 菜单 .....	15
1.11 Option 菜单 .....	16
1.12 Window 菜单 .....	17
1.13 Help 菜单 .....	18
1.14 热键 .....	19
第二章 C语言 .....	23
2.1 C语言的优点 .....	23
2.2 C语言的缺点 .....	23
2.3 C程序设计结构 .....	23
2.4 Borland C++和ANSI C .....	29
第三章 面向对象的程序设计 .....	34
3.1 封装.....	34
3.2 继承.....	35
3.3 多态性.....	38
第四章 C++语言 .....	40
4.1 C++对C语言的扩充 .....	40
4.2 C++运行库函数 .....	41

## 第二部分 Borland C++参考指南

第五章 字符分类函数 .....	43
5.1 Borland C++字符分类宏参考指南 .....	47
第六章 数据转换函数 .....	54
6.1 Borland C++数据转换函数参考指南 .....	55

第七章	目录控制函数 .....	70
7.1	Borland C++目录控制函数参考指南 .....	71
第八章	图形函数 .....	92
8.1	图形模式 .....	92
8.2	坐标系统 .....	94
8.3	图形光标 .....	94
8.4	颜色和调色板 .....	94
8.5	填充调色板 .....	96
8.6	线类型 .....	97
8.7	文本和图形的混合 .....	98
8.8	图形函数的分类 .....	100
8.9	Borland C++图形函数参考指南 .....	103
第九章	输入/输出函数 .....	155
9.1	流 I/O .....	155
9.2	低级 I/O .....	155
9.3	控制台 I/O .....	155
9.4	Borland C++ I/O 函数参考指南 .....	159
第十章	DOS、BIOS 和 8086 的接口函数 .....	236
10.1	Borland C++的 DOS、BIOS 和 8086 接口函数参考指南 .....	247
第十一章	存储器操作和字符串操作函数 .....	303
11.1	存储器操作 .....	303
11.2	字符串操作 .....	
11.3	Borland C++存储器操作和字符串操作函数参考指南 .....	306
第十二章	数学函数 .....	344
12.1	整数运算 .....	347
12.2	浮点数运算 .....	347
12.3	复数运算 .....	348
12.4	BCD 运算 .....	349
12.5	Borland C++数学函数参考指南 .....	349
第十三章	动态存储器分配函数 .....	406
13.1	段和偏移量 .....	407
13.2	一般的存储器布局 .....	407
13.3	Borland C++动态存储器分配函数参考指南 .....	408
第十四章	进程控制函数 .....	426
14.1	Borland C++进程控制函数参考指南 .....	427
第十五章	文本窗口和显示函数 .....	448
15.1	文本模式 .....	448
15.2	Borland C++文本窗口和显示函数参考指南 .....	449
第十六章	标准函数 .....	461
16.1	Borland C++标准函数参考指南 .....	462

第十七章	时间和日期函数.....	485
17.1	Borland C++时间和日期函数参考指南.....	486
第十八章	其它函数.....	502
18.1	Borland C++其它函数参考指南.....	502
第十九章	运行库函数应用例子.....	509
<b>第三部分</b>	<b>Borland C++程序设计工具</b>	
第二十章	. Turbo Assembler .....	829
20.1	处理器指令.....	803
20.2	协处理器指令.....	837
20.3	伪指令.....	839
20.4	操作符.....	846
20.5	预定义符号.....	848
第二十一章	Turbo Debugger .....	849
21.1	菜单.....	850
21.2	File 菜单 .....	850
21.3	Edit 菜单.....	852
21.4	View 菜单 .....	852
21.5	Run 菜单.....	854
21.6	Breakpoint 菜单 .....	854
21.7	Data 菜单 .....	855
21.8	Options 菜单 .....	856
21.9	Window 菜单 .....	857
21.10	Help 菜单 .....	858
21.11	Turbo Debugger 热键 .....	859
第二十二章	Turbo Profiler .....	862
22.1	菜单.....	862
22.2	File 菜单 .....	863
22.3	View 菜单 .....	863
22.4	Run 菜单.....	865
22.5	Statistics 菜单.....	865
22.6	Print 菜单 .....	867
22.7	Options 菜单 .....	867
22.8	Window 菜单 .....	867
22.9	Help 菜单 .....	869
22.10	Turbo Profiler 的热键 .....	870
<b>Borland C++3.1</b>	<b>运行库函数索引</b> .....	872

## 第一章 Borland C++ 引言

Borland C++ 提供了使用美国国家标准委员会 (ANSI) C 标准 (AT&T C++ V2.0 和 Kernighan 与 Ritchie 确定) 设计 C 和 C++ 程序的功能。C 和 C++ 程序可以在命令行或集成开发环境 (IDE) 中进行编译, 本章讨论 IDE 的组成并提供了命令行编译和连接的信息。

### 1.1 命令行编译器

虽然 Borland 提供了程序开发的 IDE, 有些用户更愿意使用命令行编译器, 因为它提供了编译过程的更直接的控制。Borland 提供的命令行编译器 BCC 自动编译和连接指定的文件, 如果需要编译 ASM 源文件则调用 Turbo Assembler 即 TASM。

命令行编译器十分灵活, Borland 提供了很多任选项控制编译过程, 如下格式用于调用命令行编译器:

```
BCC[option[option...]] [filename[filename...]]
```

这里:

option = 任选的编译器任选项, 可以没有也可以有几个任选项。

filename = 编译和连接的文件名

表 1.1 列出了命令行编译器的任选项

表 1.1 命令行编译器任选项

任选项	意 义
@ filename	响应文件
+ filename	告诉 BCC 使用另一个配置文件 filename
-A	仅使用 ANSI 关键词
-A- 或 -AT	使用 Borland c++ 关键词 (缺省)
-AK	仅使用 Kernighan 和 Ritchie 关键词
-AU	仅使用 UNIX 关键词
-a	以字对齐
-a-	以字节对齐
-B	编译时调用汇编程序处理在线宏代码
-b	使枚举为字大小 (缺省)
-b-	设置枚举类型为 signed 或 unsigned
-C	置注释嵌套 (Nested Comments) 为 on
-C-	不允许注释嵌套 (Nested Comments 为 off)
-Dname	为由空字符组成的字符串定义名



任选项	意 义
-Dname=string	为串定义名
-d	置合并相同串(Merge duplicate string)为 on
-d-	置合并相同串为 off(缺省)
-Efilename	用 filename 作为要使用的汇编程序
-efilename	连接产生 filename. EXE
-Fc	生成 COMDEFS
-Ff	自动创建远调用变量
-Ff=size	自动创建远调用变量并设置临界值
-FN	选中-FC、Ff 和 -Fs 任选项
--FS	对所有存贮模式设置 DS=SS
-f	仿浮点操作(缺省)
-f-	不做浮点操作
--f87	使用 8087 硬件指令
-f287	使用 80287 硬件指令
-ff	快速浮点操作(缺省)
--ff-	精确 ANSI 浮点操作
-G	速度优化
-G-	代码大小优化
-gn	警告:n 条信息之后停止
-H	使编译器产生并使用预编译头
-H-	不产生和使用预编译头(缺省)
-Hu	使用但不产生预编译头
-H=filename	指定预编译头文件名
-h	使用快速巨型指针算法
--IPathname	包含文件目录
-in	设有效标识符长度为 n
-Jg	生成所有模板实例的定义,并合并重复定义(缺省)
-Jgd	生成所有模板实例的公共定义,对重复定义将提示重复定义错误
--Jgx	为每个模板实例产生外部引用
-Jn	出现 n 个错误后停止编译
-K	缺省的字符类型为 unsigned

任选项	意 义
-k	生成标准堆栈框架代码
-K-	缺省的字符类型为 signed(缺省)
-Lpathname	指定库文件查找目录
-lx	将链接器命令行任选项 x 传递给链接器
-l-x	关闭链接器命令行任选项 x
-M	告诉链接器生成 .MAP 文件
-M-	告诉链接器不生成 .MAP 文件
-mc	使用紧缩(compact)模式进行编译
-mh	使用巨型(huge)模式进行编译
-ml	使用大(large)模式进行编译
-mm	使用中(medium)模式进行编译
-mm!	使用中模式进行编译并假设 DS! = SS
-ms	使用小(small)模式进行编译
-ms!	使用小模式进行编译并假设 DS! = SS
-mt	使用微小(tiny)模式进行编译
-mt!	使用缩小模式进行编译并假设 DS! = SS
-N	检查堆栈溢出
-npathname	指定输出文件的目录
-O1	生成最小的代码
-O2	生成最快的代码
-Od	关闭优化
-On	优化任选项(Oa-Ox)

任选项	意 义
-ofilename	把源文件编译成 filename.obj
-P	执行一个 C++ 编译器
-Pext	执行一个 C++ 编译器, 设置缺省扩展名为 ext 指定的扩展名
--P--	根据源文件的扩展名执行一个 C++ 或 C 编译器(缺省)
--P--ext	根据源文件的扩展名执行一个 C++ 或 C 编译器, 设置缺省扩展名为由 ext 指定的扩展名
-P	使用 pascal 语言的调用方式
--Pr	在寄存器中传递参数时使用快速调用方式
--P--	使用 C 语言调用方式(缺省)
-Qe	使用所有可用的 EMS 内存
-Qe-	不使用任何 EMS 内存
--Qx	使用所有可用的扩充内存
-r	使用寄存器变量(缺省)
--r--	不使用寄存器变量
-rd	只有用关键词 register 定义过的变量才能作为寄存器变量使用
-R	产生 ObjectBrowser 信息
-S	编译生成 .ASM 源文件而不是 .OBJ 文件
-Tstring	将命令行选择项 string 传递给 TASM 或其它用 -E 指定的汇编程序
--T--	取消所有先前给定的汇编程序命令行任选项
--tDe	生成一个 DOS .EXE 文件
-tDc	生成一个 DOS .COM 文件
-tW	生成一个与使用任选项 -W 相同的窗口模块
--Uname	取消先前用 -D 给出的宏 name 的任何定义

任选项	意 义
-u	全局标识符之前加下划线(_)
-u-	全局标识符之前不加下划线(_)
-V	带调试信息编译,且不允许内部函数扩展
--Va	通过一个临时变量的指针传递类参量
--Vb	使虚拟基类指针与基类指针的"this"指针大小相同
--Vc	不向带指向虚拟基类成员的指针的类加入隐含成员或代码
--Vf	远调用 C++ 虚表
--Vmv	对成员指针无限制
--Vmm	对成员支持多重继承
--Vms	成员指针支持单重继承
--Vmd	成员指针使用最小表达方法
--Vmp	严格按声明精度处理所有成员指针类型
--Vo	使所有"向后兼容"的 -V 开关选项有效(-Va, -Vb, -Vc, -Vp, -Vt, -Vv)
--Vp	将"this"指针作为第一个参数从堆栈传递给"pascal"成员函数
--Vs	局部 C++ 虚表
--Vt	将虚拟表指针放在非静态数据成员之后
--Vv	不改变类的布局以放松对成员指针的限制
--Vo, -Vl	外部和公共 C++ 虚表
-vi, -vi-	控制在线函数的扩充
-v, -v-	把源文件调试(source debugging)置为 on
-W	建立一个带所有可输出函数的 Windows 的 .OBJ 文件
-WD	建立一个带所有可输出函数的将被作为 .DLL(动态链接库)来连接的 Windows 的 .OBJ 文件

任选项	意 义
-WDE	建立一个带显式可输出函数的将被作为. DLL(动态连接库)来连接的 Windows 的. OBJ 文件
-WE	建立一个带显式可输出函数的 Windows 的. OBJ 文件
-WS	建立一个使用 smart 回调方式和 Windows 的. OBJ 文件
-W	显示警告信息
-Wxxx	允许 xxx 警告信息
-W-xxx	不允许 xxx 警告信息
-X	显示编译器自依赖性输出
-Y	生成覆盖代码
-Yo	覆盖被编译的文件
-y	加入行号
-Z	优化寄存器使用,禁止变量重复装入寄存器
-zAname	代码类
-zBname	BSS 类
-zCname	代码段
-zDname	BSS 段
-zEname	远(far)段
-zFname	远(far)类
-zGname	BSS 组
-zHname	远(far)组
-zPname	代码组
--zRname	数据段
-zSname	数据组

任选项	意 义
-zTname	数据类
-zX *	使用缺省名 X(缺省)
-1	生成 80186 指令
-1-	生成 8088/8086 指令和 80286 实模式指令(缺省)
-2	生成 80286 保护模式兼容指令

如下是命令行编译器的一个命令行的例子:

```
BCC -f87 -G -ml -erunprog File1 File2 File3
```

读该命令行使编译器把 File1、File2 和 File3 编译成 .OBJ 文件,然后连接它们生成一个名称为 RUNPROG.EXE 的可执行文件,采用的方式是使用 8087 硬件指令、快速优化和大内存模式。

## 1.2 IDE

该集成环境允许我们产生、编辑、编译、连接、执行和调试程序,IDE 功能强大且使用灵活,但只能在保护模式下运行。本章其余部分讨论菜单和 IDE 任选项的信息。

可以在 DOS 提示符下输入 BC 来调用 IDE,调用 IDE 的格式如下:

```
BC[option[option...]][sourcename | Projectname [sourcename]]
```

这里:

option = 一个命令行编译器任选项

sourcename Projectname = ASCII 或工程文件的名称。

正如调用 IDE 的格式指出的,命令行任选项允许我们定制该环境,表 1.2 列出了 IDE 的命令任选项。

1.2 IDE 的命令任选项

任选项	意 义
/b	重新编译和连接工程中的所有文件,在标准输出设备上打印编译信息,然后返回到操作系统
/d	如果检测到适合的硬件,该任选项使 Borland C++ 以双监视器方式运行,否则忽略该任选项
/e	该任选项指示在分配内存时把数据交换到扩充内存中
/h	产生一张所有可用命令任选项的列表
/l	如果在液晶(LCD)屏幕上运行 Borland C++ 时使用该任选项
/m	该任选项允许用户制作操作而不是构造操作(即只编译连接工程中过时的源程序文件)

任选项	意 义
/p	该任选项控制 EGA 视频适配器的调色板数据交换, 每当屏幕进行数据交换时都要恢复 EGA 调色板
/rx	指定交换驱动程序
/s	该任选项使用时编译器允许在编译时使用所有空闲内存都可被编译程序的内部表格所使用
/x	该任选项在分配内存时交换数据到扩充内存

IDE 是菜单驱动的, 很容易使用, IDE 的条形菜单 Menu 有 11 个菜单, 每个菜单提供了几个选择, 下面几节讨论了这些菜单上各任选项的信息。

### 1.3 ≡或系统菜单

≡或系统菜单(如图 1.1 所示)提供如下选择: Repaint desktop, GREP, Turbo Assembler, Turbo Debugger, Turbo Profiler, Resource Compiler 和 Import Librarian。系统菜单并不包含所有这些列出的菜单项, 只有我们已安装了有关的程序(Turbo Assembler, Turbo Profiler 等等)它们才列在菜单上, 表 1.3 描述了这些任选项。

表 1.3 三菜单的任选项

任选项	作 用
Repaint desktop	使 Borland C++ 重画屏幕
GREP	切换到 GREP
Turbo Assembler	切换到 Turbo Assembler
Turbo Debugger	切换到 Turbo Debugger
Turbo Profiler	切换到 Turbo Profiler
Resource Compiler	切换到 Resource Compiler
Import Librarian	切换到 Import Librarian

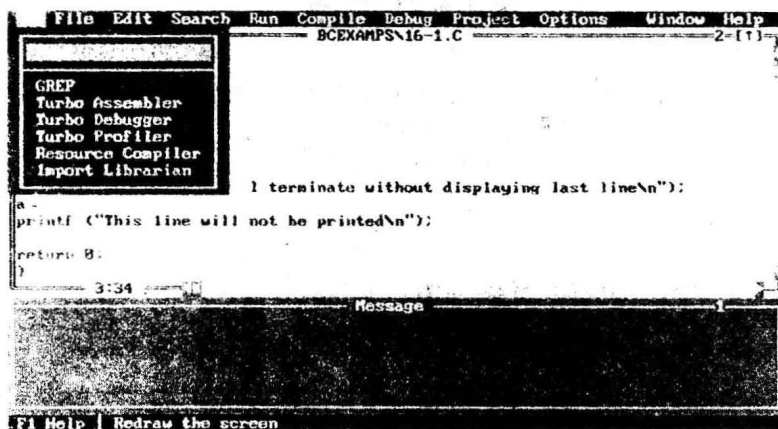


图 1.1 系统菜单

### 1.4 File 菜单

File 菜单(如图 1.2 所示)提供了几个操作文件的任选项,另外,我们可以退出 IDE 并从菜单内部得到一个 DOS 提示符,File 菜单下的任选项有 New,Open,Save,Save as,Save all,change.dir,Print,Dos shell 和 Quit,这些任选项的描述如表 1.4 所列。

表 1.4 File 菜单的任选项

任选项	作用
New	允许我们生成一个名称为 NONAME xx.C 的新文件
Open	允许我们从文件选择对话框中打开一个现存文件或者用输入到输入框的文件名建立一个新文件
Save	存贮活动窗口中的文件
Save as	存贮活动窗口中的文件到一个不同文件名中,如果可能,可以使用不同驱动器或路径
Save all	存贮打开窗口中所有修改的文件
Change dir	允许我们指定当前驱动器和路径
Print	打印活动编辑输出或信息窗口中的内容
DOS shell	允许用户暂时离开 Borland C++ 进入 DOS 以执行 DOS 命令或运行一个程序,可键入 EXIT 并按 Enter 返回 Borland C++
Quit	退出 IDE 并将其从内存中清除,如果用户没有保存所做的修改,IDE 在退出之前将询问用户是否存盘,退出后返回到 DOS 提示符

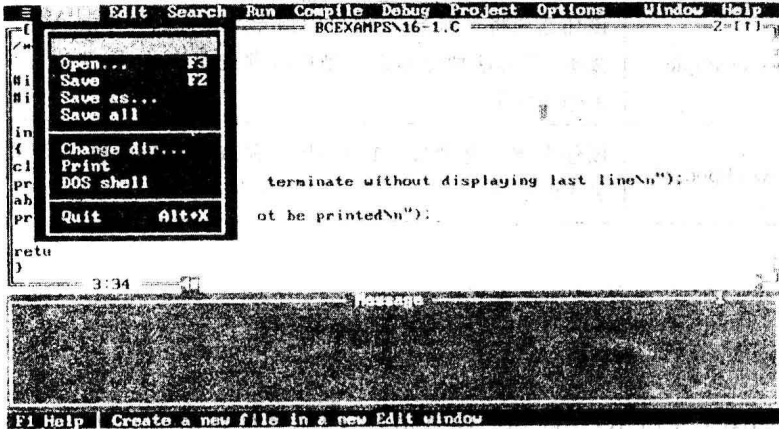


图 1.2 File 菜单



## 1.5 Edit 菜单

如图 1.3 所示说明了 Edit 菜单,它允许我们删除、复制和插入编辑窗口中或编辑窗口之间的正文,Edit 菜单有 8 个任选项:Undo、Redo、Cut、Copy、Paste、Clear、Copy example 和 Show clipboard,这些任选项的描述如表 1.5 所列。

表 1.5 Edit 菜单的任选项

任选项	作用
Undo	该任选项把当前窗口中的文件恢复成上一次编辑操作或光标移动前的原样,如果用户连续选择 Undo 命令,编辑器将反复进行翻转操作直到用户文件回到开始其当前编辑操作时的状态。
Redo	该任选项取消最后一次 Undo 命令的作用,只有紧接着一个 Undo 命令或另一个 Redo 命令的 Redo 命令才是有效的,一连串的 Redo 命令取消一连串的 Undo 命令的作用。
Cut	该任选项把选择的正文从文件中删除并放入到剪接板(clipboard),然后用户可以选取 Paste 把该正文粘贴到另一文件中或同一文件的另一地方,该段正文保持其选择状态,用户可以粘贴该正文任意多次。
Copy	该任选项把选择的正文拷贝到剪接板,然后用户可以选取 Paste 命令进行粘贴。
Paste	该任选项将正文从剪接板插入到当前编辑窗口的当前光标位置处。
Clear	该任选项删除选择的正文但不拷贝到剪接板,这样用户即使选择了 Cut 或 Copy 也不能粘贴该正文,被删除的正文是不可恢复的,除非使用 Undo 命令,如果用户想删除正文而不想覆盖剪接板中的内容时,可以选择该任选项。
Copy example	该任选项将当前 Help 窗口中预先选择好的样本正文拷入剪接板,因为这种样本正文是可以预先定义的,因此用户不必去作块标记。
Show clipboard	该任选项打开剪接板窗口,显示存放的从其它窗口剪裁、拷贝来的正文。

## 1.6 Search 菜单

Search 菜单(如图 1.4 所示)允许我们搜索正文,函数宣称和错误位置。Search 菜单下的任选项有 Find、Replace、Search again、Go to line number、Previous error、Next error 和 Locate function,这些任选项的描述如表 1.6 所列。