

Borland C++ & Turbo C++

# 库函数参考手册

李振格 主编

北京航空航天大学出版社

Borland C++ & Turbo C++

# 库函数参考手册

李振格 主编

北京航空航天大学出版社

(京)新登字 166 号

## 内 容 简 介

Borland C++和 Turbo C++支持多重覆盖窗口和自动覆盖管理,即是面向对象的程序设计语言,又是一个编辑、编译、调试、运行和剖视优化一体化的优秀的语言开发环境,且与 Turbo C 相兼容。《用户手册》、《程序员手册》和《库函数参考手册》是上述两软件的最新参考系列书。《库函数参考手册》介绍了库函数的分类, C 的主函数(main)的设计和注意事项,并按字母顺序介绍了 Borland C++ & Turbo C++的每个库函数,它们的功能、用法、原型所在的头文件、返回值、相关函数和应用范例等信息,在第二章和第四章中介绍了类库和全局变量的参考和用法。书中每个库函数的说明都附有范例。

本套书中的所有程序都通过调试。

- 书 名: Borland C++ & Turbo C++库函数参考手册
- 主 编: 李 拯 格
- 责任编辑: 肖 芝 中
- 出版者: 北京航空航天大学出版社
- 印刷者: 朝阳科普印刷厂
- 发 行 者: 新华书店总店科技发行所发行 各地新华书店经售
- 开 本: 787X1092 1/32
- 印 张: 31.25
- 字 数: 800 千字
- 印 数: 1-5000 册
- 版 次: 1992 年 2 月 第一版
- 印 次: 1992 年 2 月 第一次印刷
- 书 号: ISBN 7-81012-299-1/TP·062
- 定 价: 24.00 元

## 前 言

在 80 年代末期, 一种新的程序设计方法开始形成, 这就是所谓面向对象的程序设计。面向对象程序设计(Object Oriented Programming, 简称 OOP)吸收了结构化程序设计的所有优点, 并为程序设计者提供了分析和解决应用问题的一种新的方法。C++ 是 C 的面向对象版本, C 素以效率高、功能丰富及代码精练而著称。C++ 保留了 C 的这些重要性质, 增加了面向对象程序设计的特征。

1988 年 Borland 公司在深受欢迎的 Turbo C 的基础上, 推出最新的面向对象的程序设计软件包——Borland C++ 2.0 和 Turbo C++ 1.x。它们继承并发扬了原来 Turbo C 集成环境的优良特性, 包含了面向对象的设计和思想方法, 是目前国际上最受欢迎的面向对象程序设计软件包。Borland C++ 还可以为广为流行的 Windows 3.0 编写应用程序。

对编程者来说, 从使用一般的 C 语言到使用 Turbo C 语言, 是从一种单一的语言, 转向编辑、编译、调试和运行一体化的具有联机帮助和热键触发的集成式的语言开发环境; Borland C++ 和 Turbo C++ 的环境支持鼠标器、支持多重覆盖窗口、支持自动管理覆盖、支持内部汇编码, 集编辑、编译、调试、运行和剖视优化等功能于一体, 并具有联机帮助和热键触发的特点。因此从使用 Turbo C 语言到使用 Borland C++ 或 Turbo C++ 语言, 是从优良的环境转向优秀的环境, 是从一般的模块化程序设计转向面向对象程序设计, 标志着程序设计方法的一次飞跃。

由于 Borland C++ 和 Turbo C++ 从 Turbo C 发展而来, Borland C++ 和 Turbo C++ 都与 Turbo C 相兼容, 因此用 Turbo C 编写的程序很容易移植到 Borland C++ 或 Turbo C++ 中来。Borland C++ 和 Turbo C++ 的库函数和集成环境都与 Turbo C 很相似或者说几乎相同, 因此在本套书的编写过程中, 许多章节都先介绍 Turbo C, 然后再介绍 Borland C++ 和 Turbo C++。这样, 一方面可以让 Turbo C 程序员能很快掌握 Borland C++ 或 Turbo C++, 另一方面读者也可把本套书作为 Turbo C 1.x 和 Turbo C 2.0 的《用户手册》、《库函数手册》和《程序员手册》。

《用户手册》、《程序员手册》和《库函数参考手册》是在参考 Borland C++ 和 Turbo C++ 两个软件的所有联机资料和许多关于 Borland C++ 和 Turbo C++ 程序设计方法的书籍后编写而成的, 是掌握和使用 Borland C++ 和 Turbo C++, 编写 C++ 程序的不可缺少的书籍。

《用户手册》介绍了 Borland C++ 和 Turbo C++ 软件的安装步骤, 说明了集成环境、实用程序和资源管理程序的使用方法, 描述了编辑器宏语言, 指出了 Microsoft 程

序员使用 Borland C++ 和 Turbo C++ 的注意事项。其中着重说明了集成环境编辑器、调试器和项目管理程序的用法。

《程序员手册》介绍了 C++ 语言的所有语法成分，提供了 C++ 的关于内存模式、混合模型编程、流 I/O、视频函数、浮点数处理、覆盖、编写 Windows 3.0 应用程序和错误信息的技术参考资料，用示例说明了利用 Turbo C、Borland C++ 和 Turbo C++ 先进技术进行程序设计的方法。在书中还通过丰富的范例由浅入深地讨论了类、对象、层次、重载、封装、继承、虚函数和多态性等概念，说明了利用 C++ 进行面向对象程序设计的方法。

《库函数参考手册》介绍了库函数的分类，C 的主函数(main)的设计和注意事项，并按字母顺序介绍了 Borland C++ 和 Turbo C++ 的每个库函数，它们的功能、用法、原型所在的头文件、返回值、相关函数和应用范例等信息，在第二章和第四章中介绍了类库和全局变量的参考和用法。书中每个库函数的说明都附有范例。

本套书中的所有程序都通过调试。

虽然在国外，特别在美国，面向对象程序设计方法已发展到实用的水平，许多软件公司都已采用 C++ 语言编写软件，但在国内，面向对象的概念还处于理论的初级阶段，许多概念和术语都没有统一，有些概念还没有明确，加上编者水平有限，书中不妥之处恳请广大读者指正。

本书在编写过程中，得到了许多同志的帮助和支持，毕卫国参加了本套书中部分章节的编写工作。章忆文、汪亚文、刘云、朱海斌、刘晓东、吕良双、李兵、黄磊光、张军也帮助整理了一些文稿。

在出版的过程中，得到了杨昌竹、肖之中、樊毅等老师的许多有益的指导，在此表示衷心的感谢。

最后特别感谢大力举荐本套书的蔡德林老师和慧眼选中本套书的杨昌竹老师，正是他们才使得本套书有一个与读者见面的机会。

李振格

1991年6月于北航

# 目 录

## 前 言

<b>第一章 运行时间库分类</b> .....	<b>1</b>
1.1 为何要存取运行时间库源代码.....	1
1.2 Borland C++ & Turbo C++ 头文件.....	2
1.3 库子程序分类.....	3
1.3.1 分类子程序.....	3
1.3.2 转换子程序.....	3
1.3.3 目录控制子程序.....	4
1.3.4 诊断子程序.....	4
1.3.5 图形子程序.....	4
1.3.6 输入/输出子程序.....	5
1.3.7 接口子程序(DOS、8086 和 BIOS).....	6
1.3.8 操作子程序.....	7
1.3.9 数学子程序.....	7
1.3.10 存储子程序.....	8
1.3.11 杂类子程序.....	9
1.3.12 进程控制子程序.....	9
1.3.13 标准子程序.....	9
1.3.14 文本窗口显示子程序.....	9
1.3.15 时间和日期子程序.....	10
1.3.16 变量参数表子程序.....	10
<b>第二章 类 库</b> .....	<b>11</b>
2.1 类库概述及类的重要概念.....	11
2.1.1 基于对象层次的抽象类.....	12
2.1.1.1 Container 类.....	12
2.1.1.2 Collection 层次.....	12
2.1.1.3 ContainerIterator 层次.....	13
2.2 类库中所使用的约定.....	14
2.2.1 文件命名.....	14
2.2.2 源命名.....	14
2.2.3 注释风格.....	14
2.2.3.1 模块头.....	15
2.2.3.2 依赖性.....	15
2.2.3.3 类型头.....	15
2.2.3.4 函数头.....	16
2.2.4 类库目录.....	17
2.2.5 INCLUDE 目录.....	17
2.2.6 SOURCE 目录.....	17
2.2.7 LIB 目录.....	17
2.2.8 EXAMPLES 目录.....	17
2.3 类调用.....	17
AbstractArray.....	18
Array.....	19
ArrayIterator.....	20
Association.....	21
Bag.....	22

BaseData	23
BaseTime	24
Collection	25
Container	26
ContainerIterator	27
Date	27
Dictionary	27
Deque	28
DoubleList	30
DoubleListelement	31
DoubleListIterator	31
Error	32
HashTable	32
HashTableIterator	33
List	34
ListElement	34
ListIterator	35
Object	35
Queue	36
Set	38
Sortable	38
SortedArray	39
Stack	40
String	41
time	43
<b>第三章 运行库函数参考</b>	<b>44</b>
3.1 main——主函数	44
3.1.1 main 参数	44
3.1.2 使用 P(Pascal 调用约定)	47
3.1.3 main 返回值	47
3.2 库参考条目	47
abort 异常终止—进程	48
abs 返回整数的绝对值	48
absread 读磁盘的绝对扇区	49
abswrite 写磁盘绝对扇区	49
access 确定文件的存取权限	50
acos 计算反余弦值	51
allocmem 分配 DOS 内存	52
arc 画圆弧	52
arg 求复平面中一个复数的弧度	54
asctime 转换日期和时间为对应的 ASCII 码	54
asin 反正弦函数	55
assert 条件终止函数	56
atan 反正切函数	57
atan2 计算 y/x 的反正切值	57
atexit 定义终止函数	58
atof 将字符串转换成浮点数	58
atoi 把字符串转换成整数	59
atol 把字符串转换成整型	60
bar 画二维条形图	61
bar3d 画一个三维条形图	62

bcd	把一个数转换成对应的BCD码	63
bdos	DOS系统调用	64
bdosptr	DOS系统调用	64
bioscom	I/O通信	66
biosdisk	调用BIOS磁盘驱动程序	67
biosequip	检查设备	70
bioskey	调用BIOS的键盘接口	71
biosmemory	返回内存大小	72
biosprint	调用BIOS的打印机I/O接口	73
biostime	读取或设置BIOS时钟	74
brk	改变数据段内存分配	75
bsearch	数组的二分法搜索	75
cabs	计算复数的模	77
calloc	分配内存	77
ceil	舍入	78
cgets	读字符串	79
chdir	改变当前目录	79
_chmod	改变文件的存取权限	81
chmod	改变文件存取权限	82
chsize	修改文件长度	83
circle	画圆	84
_clear87	清除浮点状态字	85
cleardevice	清图形屏幕	85
clearerr	复位错误标志	86
clearviewport	清除当前图形窗口	87
clock	测定运行时间	88
_close	关闭文件	89
close	关闭文件	90
closegraph	关闭图形系统	91
clreol	清除从当前光标位置到行尾的字符	92
clrscr	清除文本窗口,并把光标放在左上角	92
complex	创建复数	93
conj	求复数的共轭复数	94
_control87	处理浮点控制字	94
coreleft	返回尚未使用的内存(RAM)大小	95
cos	计算余弦值	96
cosh	计算双曲余弦值	96
country	读取与特定国家有关的格式	97
cprintf	格式化并输出数据至屏幕	98
cputs	输出一字符串至屏幕	99
_creat	创建一个新文件或重写一个已存在的文件	99
creat	创建一个新文件或重写一个已存在的文件	100
creatnew	创建新文件	101
createmp	创建一个文件名唯一的文件	102
cscanf	从控制台执行格式化输入	103
ctime	把日期和时间转化为对应的字符串	104
ctrlbrk	设置ctrl-break处理程序	104
delay	暂停	105
delline	在文本窗口中删去一行	106
detectgraph	检测硬件并确定应使用何种图形驱动程序和图形模式	106
difftime	计算两个时刻之间的时间差	109



disable	屏蔽中断	109
div	将两个整数相除, 返回商和余数	110
dosexterr	获取扩展错误信息	111
dostounix	把日期和时间转换成 UNIX 格式	112
drawpoly	绘制多边形	112
dup	复制文件句柄	114
dup2	将一个文件句柄(oldhandle)复制到一个已有的文件句柄(newhandle)	115
ecvt	把浮点数转换为字符串	116
ellipse	绘制椭圆	117
_emit	将文字值直接插入源程序中	118
enable	开硬件中断	119
eof	检测文件是否结束	120
execl, execl, execlp, execlpe		
execv, execve, execvp, execvpe	装入并运行其它程序	121
_exit	终止程序	125
exit	终止程序	125
exp	计算 e 的 x 次方	126
fabs	返回浮点数的绝对值	127
faralloc	从远程堆中分配内存	127
farcoreleft	返回远程堆中未使用内存的大小	128
farfree	从远程堆中释放一块已分配内存	128
farheapcheck	校验远程堆中分配的内存块	129
farheapcheckfree	检查远程堆中未分配的内存	130
farheapchecknode	检测并证实远程堆上的某个节点	131
farheapfillfree	将一个常量填充在远程堆上的空闲内存中	133
farheapwalk	按节点顺序逐个查询远程堆	134
farmalloc	从远堆中分配内存	135
farrealloc	调整远堆中的已分配块	136
fclose	关闭一个流	136
fcloseall	关闭打开流	137
fcvt	将浮点数转换为字符串	138
fdopen	把流与一个文件句柄相联	138
feof	检测流上的文件结束标志	140
ferror	检测流上的错误	140
fflush	刷新一个流	141
fgetc	从流中读取字符	142
fgetchar	从流中读取字符	143
fgetpos	取得当前文件指针	143
fgets	从流中读取一字符串	144
filelength	取文件长度	145
fileno	取得文件句柄	145
fillellipse	画椭圆饼	146
fillpoly	画多边形	146
findfirst	查找第一个匹配文件	148
findnext	查找下一个匹配文件	149
floodfill	填充区域	150
floor	下舍入	151
flushall	刷新所有流	152
fmod	计算 x/y 的余数	153
fnmerge	建立文件路径	153
fnsplit	分解完整的路径名	154

fopen	打开一个流	155
FP_OFF	获取远地址偏移量	157
_fpreset	重新初始化浮点数学包	158
fprintf	传送输出到一个流中	159
FP_SEG	获取远地址段值	160
fputc	送一个字符到一个流中	161
fputchar	送一个字符到标准输出	161
fputs	送一个字符串到流中	162
fread	从流中读数据	162
free	释放已分配的内存	163
freemem	释放先前分配的 DOS 内存	163
freopen	把一个新文件同个打开的流相联	164
frexp	对双精度数进行科学计数	165
fscanf	格式化输入	166
fseek	移动文件指针	166
fsetpos	定位文件指针	167
fstat	获取已打开文件的信息	169
ftell	返回当前文件指针	170
ftime	把当前时间存入 timeb 结构中	171
fwrite	把参数写入流中	172
gcvt	把浮点数转换为字符串	172
geninterrupt	产生软中断	173
getarccorrd	取得最后一次调用 arc 的坐标	174
getaspectratio	返回当前图形模式的纵横比	176
getbkcolor	返回当前背景颜色	177
getc	从流中取字符	178
getcbrk	获取 control-break 状态	179
getch	从键盘无回显地读取一字符	179
getchar	从 stdin 流中读取一个字符	180
getche	从键盘并回显地读取一字符	180
getcolor	返回当前绘图颜色	181
getcurdir	读取指定驱动器的当前目录	182
getcwd	读取当前目录	183
getdate	读取系统日期	183
getdefaultpalette	返回缺省调色板信息	184
getdfree	读取磁盘空闲空间	185
getdisk	读取当前磁盘驱动器号	186
getdrivename	返回指向当前图形驱动程序名字的指针	186
getdta	读取磁盘传输地址	187
getenv	读取环境变量的当前值	188
getfat	读取指定驱动器的 FAT 信息	188
getfatd	读取驱动器 FAT 信息	190
getfillpattern	将用户定义的填充模式拷贝到内存	190
getfillsettings	取得当前填充模式和填充颜色的有关信息	192
getftime	读取文件日期和时间	194
getgraphmode	返回当前图形模式	195
getimage	将指定区域的位图象存入内存	196
getlinesettings	读取当前线型、模式和宽度	198
getmaxcolor	返回可选的最大有效颜色值	200
getmaxmode	返回当前驱动程序的最大图形模式号	201
getmaxx	返回屏幕上最大的 x 坐标值	202

getmaxy	返回屏幕上最大的 y 坐标值	203
getmodename	返回指向含有指定图形模式名字字符串的指针	204
getmoderange	获取图形驱动程序的模式范围	205
getpalette	返回当前调色板的有关信息	206
getpalettesize	返回调色板的颜色数目	208
getpass	读入口令	209
getpid	读取进程号	210
getpixel	读取得象素的颜色	210
getpsp	读取程序段前缀	212
gets	从标准输入流 stdin 中读取一字符串	212
gettext	拷贝文本屏幕上的文本拷贝到内存中	213
gettextinfo	读取文本模式的显示信息	214
gettextsettings	返回当前图形字体的有关信息	215
gettime	读取系统时间	217
getvect	读取中断向量	217
getverify	取得 DOS 的当前校验状态	218
getviewsettings	返回有关当前视区的的信息	219
getw	从输入流中读取一整数	220
getx	返回当前图形方式下位置的 x 坐标值	221
gety	返回当前位置的 y 坐标值	222
gmtime	把日期和时间转换为格林威治标准时间(GMT)	223
gotoxy	在文本窗口中定位文本光标	224
graphdefaults	复位图形设置	225
grapherrormsg	返回一个指向错误信息串的指针	226
_graphfreemem	可修改的图形内存释放函数	227
_graphgetmem	可修改的图形内存分配函数	228
graphresult	返回最后一次失败图形操作的错误码	230
Harderr	建立一个错误处理程序	231
hardresume	硬件错误处理函数	235
hardretn	硬件错误处理函数	237
heapcheck	检测堆	239
heapcheckfree	检查堆上的自由块	240
heapchecknode	检查并且验证堆上的一个节点	241
heapfillfree	用常量填充堆上的自由块	242
heapwalk	检查所有的堆	243
highvideo	选择高亮度字符	244
hypot	计算直角三角形的斜边长	245
imag	计算一个复数的虚部	245
imagesize	返回保存位图象所需的缓冲区大小	246
initgraph	初始化图形系统	248
inport	从端口中读入一个字	251
inportb	从端口中读入一个字节	251
inline	在文本窗口插入一空行	252
installuserdriver	安装设备驱动程序到 BGI 设备驱动程序表中	252
installuserfont	安装未嵌入 BGI 系统的字体文件(.CHR)	254
int86	调用 8086 软中断	255
int86x	通用 8086 软中断接口	256
intdos	通用 DOS 中断接口	257
intdosx	通用 DOS 中断接口	258
intr	改变软中断接口	259
ioctl	I/O 设备控制	260

isalnum	字符分类宏	261
isalpha	字符分类宏	262
isascii	字符分类宏	262
isatty	检查设备类型	263
isctrl	字符分类宏	264
isdigit	字符分类宏	264
isgraph	字符分类宏	265
islower	字符分类宏	265
isprint	字符分类宏	266
ispunct	字符分类宏	266
isspace	字符分类宏	267
isupper	字符分类宏	267
isxdigit	字符分类宏	268
itoa	把整数转换为字符串	268
kbhit	检查当前按下的键	269
keep	驻留并退出	269
labs	给出长型绝对值	271
ldexp	计算x乘以2的exp次方	271
ldiv	两个长整型数相除, 返回商和余数	272
lfind	线性搜索	272
line	在指定两点间画一直线	273
linerel	从当前位置(CP)到与CP有一相对距离的点画一直线	274
lineto	从当前位置到(x, y)画一直线	275
localeconv	返回指向当前 locale 结构的指针	276
localtime	把日期和时间转变为结构类型	277
lock	设置文件共享锁	278
log	计算x的自然对数	279
log10	计算log(X)	280
longjmp	执行非局部跳转	281
lowvideo	选择低亮度字符	282
_lrotl	将无符号长整型数向左循环移位	282
_lrotr	将无符号长整型数向右循环移位	283
lsearch	线性搜索	284
lseek	移动文件指针	285
ltoa	把一个长整型数转换为字符串	286
malloc	分配内存	287
matherr	用户可修改的数学错误处理程序	287
max	返回两数中较大的数	289
memcpy	拷贝一个n字节长的字符串	289
memchr	在字符串中搜索字符	290
memcmp	比较两个字符串	291
memcpy	拷贝字符串	291
memcmp	比较两个字符数组中的n个字节, 忽略大小写	292
memmove	拷贝块中的n字符	293
memset	将一个内存块的n个字节都设置为c	293
min	返回两个值中较小的一个	293
mkdir	创建目录	294
MK_FP	设置一个远指针	295
mktemp	建立一个唯一的文件名	296
mktime	把时间转换成日历形式	296
modf	把双精度数转化为科学计数法	297

movedata	拷贝数据	298
moverel	从当前位置(CP)移动一相对距离	298
movetext	将屏幕上的文本从一个矩形区域拷贝到另一个矩形区域	300
moveto	从当前坐标位置(CP)移到(x, y)	300
movmem	移动一长为 length 字节的串	302
norm	返回复数的模	302
normvideo	选择正常亮度字符	303
nosound	关闭 PC 机扬声器	303
open	打开一个文件进行读或写	304
open	打开一个文件进行读或写	305
outport	输出一个字到端口中	306
outportb	输出一个字节到端口	307
outtext	显示一个字符串	307
outtextxy	在指定位置显示一字符串	309
OvrlnitEms	初始化 EMS, 以便覆盖管理程序用于交换	310
OvrlnitExi	初始化 EXT 内存, 以便覆盖管理程序用于交换	310
parsfnm	分析文件名	311
peek	返回由 segment: offset 指定的内存中的字	312
peekb	返回由 segment: offset 指定的内存中的字节	313
perror	打印系统错误信息	314
pieslice	绘制并填充扇形	315
poke	在由 segment: offset 指定的内存中存储一个字	316
pokeb	在由 segment: offset 指定的内存中存储一个字节	317
polar	用给定幅度和角度计算复数值	317
poly	根据参数产生一个多项式	318
pow	计算 x 的 y 次方	319
pow10	指数函数 10 的 p 次方	319
printf	写格式化输出到 stdout	320
putc	输出一个字符到流中	326
putch	向屏幕输出字符	326
putchar	在 stdout 上输出字符	327
putenv	将字符串放入当前环境中	328
putimage	输出一个位图象到图形屏幕上	329
putpixel	写象素点	331
puts	输出一字符串到标准输出(stdout)	332
puttext	从内存区拷贝文本到屏幕	333
putw	输出一整数到流中	333
qsort	用快速排序算法进行排序	335
raise	向正在执行的进程发送一个软中断信号	336
rand	产生随机数	336
randbrd	随机块读	337
randbwr	随机块写	338
random	随机数发生器	340
randomize	初始化随机数发生器	340
read	读文件	341
read	读文件	342
real	返回复数的实部	344
realloc	重新分配内存	344
rectangle	画一个矩形	345
registerbgidriver	注册已加载或连接进来的图形驱动程序	346
registerbgifont	注册已连接进来的矢量字体代码	347

remove	删除一个文件	349
rename	文件改名	349
restorecrtmode	恢复屏幕为调用 initgraph 前的设置	350
rewind	将文件指针重定位于流的开始处	352
rmdir	删除目录	352
_rotl	将一个无符号整数(unsigned)左循环移位	353
_rotr	将一个无符号整数向右循环移位	354
sbrk	改变数据段地址	355
scanf	格式化输入	356
searchpath	按 DOS 路径查找一个文件	362
sector	画并填充椭圆扇区	363
segread	读段寄存器值	364
setactivepage	设置图形输出活动页	364
setallpalette	改变所有的调色板颜色	366
setaspectratio	设置图形纵横比	368
setbkcolor	用调色板设置当前背景颜色	370
setblock	修改已分配的内存的大小	371
setbuf	把缓冲区与流相联	372
setcbkr	设置 control-break	373
setcolor	设置当前要画的线的颜色	374
_setcursortype	选择光标类型	376
setdate	设置 DOS 日期	376
setdisk	设置当前驱动器	377
setdta	设置磁盘传输地址	378
setfillpattern	选择自定义的填充模式	379
setfillstyle	设置填充模式和颜色	380
setftime	取得文件日期和时间	383
setgraphbufsize	改变内部图形缓冲区的大小	384
setgraphmode	将系统设置成图形模式并清屏	385
setjmp	非局部跳转(在 MS-WINDOWS 中不能使用本功能)	387
setlinestyle	设置当前画线宽度和类型	388
setlocale	选择一个场所(该函数用于提供语言和国家信息)	390
setmem	设置内存	391
setmode	设置打开文件方式	391
setpalette	改变调色板的颜色	392
setrgbpalatte	定义 IBM 8514 图形卡的颜色	394
setttextjustify	为图形函数设置文本的对齐方式	395
setttextstyle	为图形输出设置当前的文本属性	397
settime	设置系统时间	400
setusercharsize	修改矢量字体字母的宽度和高度	400
setvbuf	使缓冲区与流相联	402
setvect	设置中断矢量入口	403
setverify	设置 DOS 中的校验标志状态	404
setviewport	为图形输出设置当前视口	405
setvisualpage	设置可见的图形页号	406
setwritemode	设置图形方式下画线的输出模式	407
signal	设置某一信号的对应动作	408
sin	计算正弦值	411
sinh	计算双曲正弦值	412
sleep	执行挂起一段时间	413
sopen	打开一共享文件	413

sound	按指定频率打开 PC 扬声器	415
spawnl, spawnl, spawnlp, spawnlpe, spawnv		
spawnve, spawnvp, spawnvpe	创建并运行子进程	416
sprintf	送格式输出到字符串	418
sqrt	计算参数平方根的绝对值	419
srand	初始化随机数发生器	419
sscanf	从某串中扫描格式化输入	420
stat	读取文件信息	422
_status87	取浮点状态	423
stime	设置系统日期和时间	424
stpcpy	拷贝字符串	425
strcat	串连接	425
strchr	搜索串中某个给定字符的第一次出现	426
strcmp	串比较	426
strcmpi	忽略大小写的串比较	427
strcoll	比较两个串	428
strcpy	串拷贝	429
strcspn	搜索串中不包含给定字符集之子集的第一个段	429
strdup	复制串	429
strerror	建立用户定义的错误信息	430
strerror	返回指向错误信息字符串的指针	431
strftime	时间的格式化输出	431
stricmp	忽略大小写的串比较	433
strlen	计算字符串的长度	433
strlwr	转换字符串中的大写字母为小写字母	434
strncat	把字符串的一部分附加到另一个串之后	434
strncmp	把串的一部分与另一个串的一部分进行比较	435
strncmpi	忽略大小写的串部分比较	436
strnset	将串中指定数目字节设置为字符	436
strpbrk	搜索给定集合中任一字符在串中的首次出现	437
strrchr	搜索给定字符在串中的最后一次出现	437
strrev	颠倒串中各字符的顺序	438
strset	设置串中所有字符为给定字符	438
strspn	搜索给定字符集的子集在串中第一次出现的段	439
strstr	搜索给定子串在某串中的出现位置	439
strtod	把串转换为双精度数值	440
strtok	搜索串中的某单词, 该单词由第二个串中指定的符号进行分隔	441
strtol	转换串为长整型数	441
strtoul	将字符串转换为给定基数的无符号长整型值	443
swab	交换字节	443
system	执行 DOS 命令	444
tan	计算正切值	444
tanh	计算参数 x 的双曲正切值	444
tell	取文件指针的当前位置	445
textattr	设置文本属性	445
textbackground	选择文本的背景颜色	446
textcolor	选择文本模式的前景颜色	448
textheight	返回以象素为单位的字符串高度	449
textmode	将屏幕设置成文本模式	450
textwidth	返回以象素为单位的字符串宽度	451
time	取时间	452
		453

tmpfile	以二进制方式打开临时文件	454
tmpnam	创建唯一的文件名	455
toascii	转换字符为 ASCII 格式	455
_tolower	转换字母为小写	456
tolower	转换字符为小写	456
_toupper	转换字母为大写	457
toupper	转换字符为大写	458
tzset	设置全局变量 daylight、timezone 和 tzname 的值	458
uitoa	转换无符号长整型值为字符串	459
ungetc	把一个字符回退到输入流中	460
ungetch	把一个字符回送到键盘缓冲区	460
unixtocms	把 UNIX 格式的日期和时间转换成 DOS 格式	461
unlink	删除文件	462
unlock	解除文件共享锁	463
va_arg	va_end、va_start 实现可变参数表	464
vfprintf	送格式化输出到一流中	465
vfscanf	从流中搜索和格式化输入	466
vprintf	送格式化输出到 stdout	467
vscanf	从 stdin 中搜索和格式化输入	468
vsprintf	送格式化输出到串中	469
vsscanf	从流中搜索和格式化输入	470
wherex	给出窗口内光标水平位置	471
wherey	给出窗口内光标垂直位置	471
window	创建活动文本模式窗口	471
_write	写文件	472
write	写文件	473
<b>第四章 全局变量</b>		<b>475</b>
_8087	协处理器芯片标志	475
_argc	保存命令行的参数个数	475
_argv	命令行参数指针数组	475
_ctype	字符属性信息数组	475
daylight	指示是否进行夏令时间调整	475
directvideo	视频输出控制的标志	476
environ	存取 DOS 环境变量	476
errno、_doserrno、sys_errlist、sys_nerr	使 perror 能打印错误信息	476
_fmode	设置缺省文件传送模式	478
_heaplen	保存近堆的长度	478
_openfd	存取模式数组	479
_osmajor、_osminor、_osminor	包含 DOS 版本的主号和次号	479
_psp	包含当前程序的程序段前缀(PSP)的段地址	479
_stklen	保存堆栈的大小	479
timezone	包含当地时间与格林威治时间(GMT)之间的差值(以秒为单位)	480
tzname	时区名指针数组	480
_version	DOS 版本号	480
_wscroll	指示控制台 I/O 函数是否滚屏	481



# 第一章 运行时间库分类

本章概述 Borland C++ & Turbo C++ 的库子程序和包含文件。

在本章中，将要

- 解释为什么要求得到 Borland C++ & Turbo C++ 运行时间库的源代码；
- 列出并描述头文件；
- 总结库子程序所执行任务的不同分类。

Borland C++ & Turbo C++ 软件带有 450 多个函数和宏，函数和宏可从 C 程序中调用，以执行各种不同的任务，包括低级 I/O 和高级 I/O、字符串和文件操作、内存分配、进程控制、数据转换、数学计算等等。这些函数和宏称为库子程序，将在第二章中逐个详尽介绍。

Borland C++ & Turbo C++ 子程序包括在库文件 (Cx.LIB、CPx.LIB、MATHx.LIB 和 GRAPHICS.LIB) 中。由于 Borland C++ & Turbo C++ 支持 6 种不同的内存模式，除了微型模式之外的其它模式都有自己的库文件和数学子程序文件，它们包含有为相应模式所写的子程序 (微型模式与小型模式共用库和数学子程序文件)。

Borland C++ & Turbo C++ 实现了最新 ANSI 标准，它允许 (并强烈推崇) 对程序中的子程序给出函数原型。每个 Borland C++ & Turbo C++ 子程序都在一个或多个头文件中用原型进行了说明。

## 1.1 为何要存取运行时间库源代码

Borland C++ & Turbo C++ 运行时间库包含 450 多个函数，它们可应用于各种不同的方面：对 IBM PC 机的低级控制、与 DOS 的接口、输入/输出、进程管理、串和内存操作、数学、排序和搜索等。为什么 Borland 公司提供这些函数的源代码呢？主要有以下几个方面的原因：

- 用户可能发现要编写的 Borland C++ & Turbo C++ 函数类似于但又不完全等同于库中函数。通过存取运行库源代码，能剪裁库函数以满足需要，而免除编写一个自己的单独函数的繁琐工作。
- 在调试代码时，用户有时希望知道有关库函数的更多细节，这时若有运行时间库的源代码，将会有很大帮助。
- 当不能明确应该调用那个库函数时，快速地查阅一下此类函数的源代码有利于用户作出判断。
- 去掉 C 符号中的开头下划线或改变其它一些写法可能更符合用户的编程习惯，若能存取运行时间库的源代码，则能做到这一点。
- 通过学习专家编写的库源代码，用户能学到许多东西。

由于这些原因或者其它的原因，希望能够存取 Borland C++ & Turbo C++ 运行时间库的源代码。另外，Borland 公司也非常信奉“开放结构”的原理，因而公开了 Borland C++ & Turbo C++ 运行时间库的源代码。