

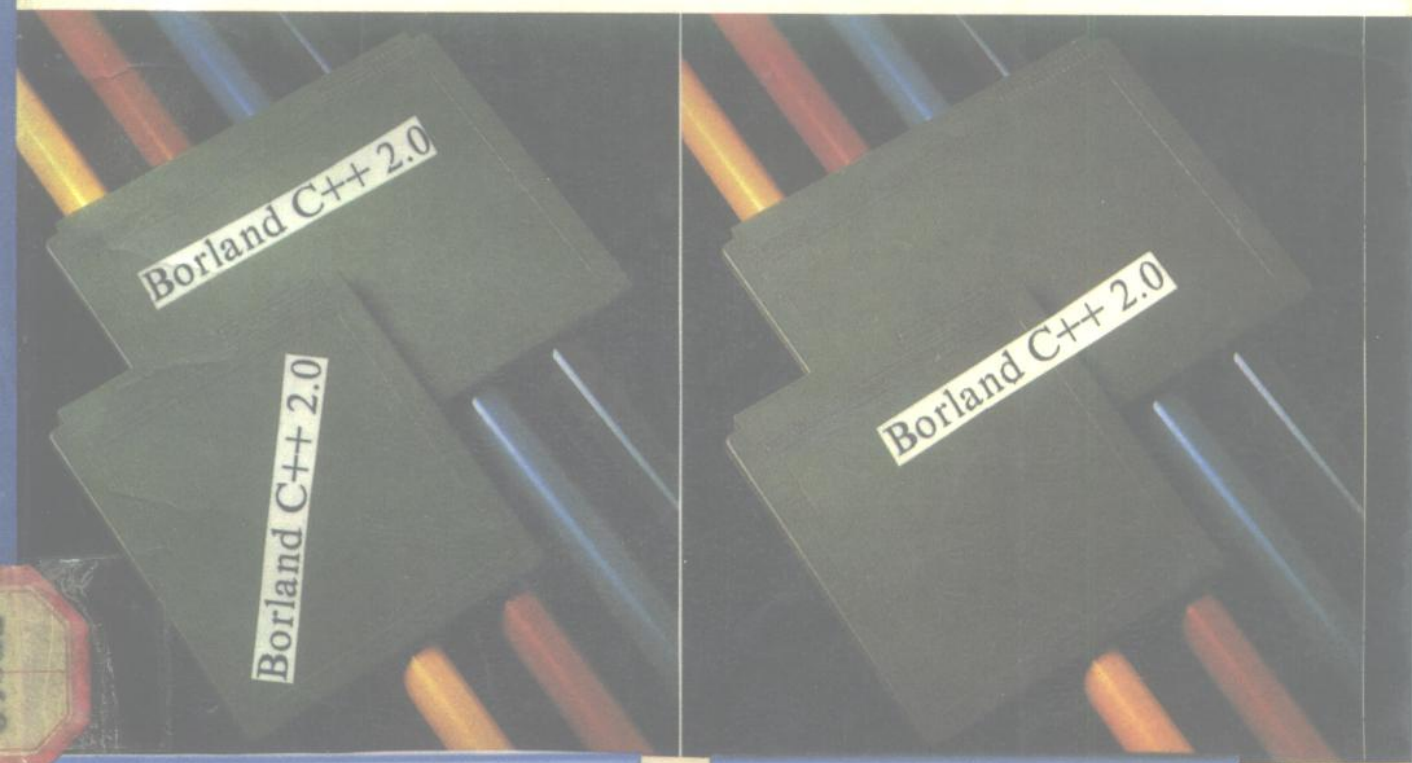
(共三册)

HOPE

Borland C++ V2.0

技 术 丛 书

■ 库函数参考手册



北京希望电脑公司

2

TP312  
C080  
2

Borland C++ 2.0技术丛书之二(共三册)

# Borland C++库函数参考手册

碧河 等编译

丛书的内容包括:

- 《入门》
- 《用户手册》
- 《程序设计手册》
- 《Whitewater 资源开发工具》

北京希望电脑公司

一九九一年八月

## 前 言

随着计算机程序设计方法的不断提高和改进，面向对象的程序设计技术在我国也逐渐流行起来，并有迅猛发展的趋势。它提出了一种全新的程序设计思想，把数据和对数据的操作融合起来。

美国 Borland 公司在 Turbo C 的基础上，曾推出过面向对象的程序设计软件包——Turbo C++。它继承并发展了原来 Turbo C 集成环境的优良特性，并包含了面向对象的基本思想和设计方法，曾是国际上最受欢迎的面向对象程序设计软件包。

去年五月，美国 Microsoft 公司推出了 Windows 3.0，立刻风靡全球，在微机软件世界引起了一革命。许多软件公司纷纷推出适用于 Windows 3.0 的软件。Borland 公司也不甘落后，于今年二月推出了适用于开发 Windows 应用程序的 C++ 版本——Borland C++ 2.0。

除了与早期版本的 Turbo C、Turbo C++ 完全兼容外，Borland C++ 2.0 还提供了以下新特色：

- C++: Borland C++ 提供 C++ 编程的全部功能 (AT&T C++ 2.0 版的实现)，并提供了一个 C++ 的类库及对 C++ 1.2 版流的支持。
- ANSI C: Borland C++ 提供至今为止最新的 ANSI C 标准的实现。
- Microsoft Windows 应用程序: 可以用 Borland C++ 编制 Windows 应用程序。Borland C++ 中添加了许多新的功能，包括资源编译器和资源开发工具。
- 予编译的头文件，可以缩短程序的编译时间。
- 实模式和保护模式的编译器。
- 类库中提供了集合、数组等类型。
- 求助系统中添加了 Windows API。
- Borland 新的程序员平台: 程序员平台是新型的用户接口，它胜过了旧的集成环境，提供对计算机上全范围内程序和工具的访问。包括：
  - 鼠标器支持
  - 多重覆盖窗口
  - 多重文件编辑
  - 支持内部汇编码
  - 内部汇编器

集成调试器

对大缓冲区的恢复 (undo) 和重复 (redo) 功能

等等。

- VROOMM (面向对象的实时虚拟存储管理): VROOMM使你能很简单地覆盖代码。
- 新的程序员平台的联机访问。
- 联机的快速文本求助, 用拷贝的模拟程序例子试验每一个函数。
- 许多独立的库函数, 包括堆检测函数和完整的复数与BCD数学函数集。

其它特征包括:

- 对-S 选择的扩展: C 源码作为注释可加到结果汇编代码上。
- 远程对象和大容量数组
- 候补.CFG文件: 用户能创建几个这样的文件, 并且在需要的时刻使用合适的一个。
- 用于命令行编译的响应文件。
- 快速完成复杂的算术运算。

我们在长期从事 Turbo C 和面向对象程序设计的基础上, 根据有关软件和资料编译了这套 Borland C++程序设计丛书。

本丛书共有五册, 分别为《入门》、《用户手册》、《程序设计手册》、《库函数参考手册》、《Whitewater 资源开发工具》, 它们全面系统地介绍了 Borland C++的基础知识和高级技术, 是一套引导读者进行面向对象程序设计和进行 Windows 应用编程的参考书。

在本丛书出版过程中, 得到中国科学院希望高级电脑技术公司资料部秦人华经理、杨淑欣老师的大力帮助和支持, 在此表示衷心的感谢。

编译者

一九九一年五月于北京

# 目录

简介 (1)

类和成员函数文档 (1)

第一章 main 函数 (3)

main 的变元 (3)

例子 (3)

通配符参数 (4)

使用 -P 选项 (Pascal 调用约定) (5)

main 的返回值 (5)

第二章 运行库 (6)

函数项说明 (6)

A

abort (7)    abs (7)    absread (8)    abswrite (9)    access (9)  
acos (10)    allocmem (10)    arc (11)    arg (12)    asctime (13)  
asin (14)    assert (14)    atan (15)    atan2 (15)    atexit (16)  
atof (17)    atoi (17)    atol (18)

B

bar (18)    bar3d (19)    bcd (20)    bdos (21)    bdosptr (22)  
bioscom (22)    biosdisk (24)    biosequip (25)    bioskey (27)  
biosmemory (28)    biosprint (28)    biostime (29)    brk (30)  
bsearch (31)

C

cabs (32)    calloc (33)    ceil (33)    cgets (34)    chdir (34)  
\_chmod (35)    chmod (37)    chsize (38)    circle (38)    \_clear87 (39)  
cleardevice (40)    clearerr (41)    clearviewport (42)    clock (43)  
\_close (43)    close (44)    closegraph (45)    creol (46)  
clrscr (46)    complex (47)    conj (47)    \_control87 (48)  
coreleft (48)    cos (49)    cosh (49)    country (50)    cprintf (50)  
cputs (51)    \_creat (51)    creat (52)    creatnew (53)  
createmp (54)    cscanf (55)    ctime (55)    ctrlbrk (55)

D

delay (56)    delline (57)    detectgraph (57)    difftime (58)  
disable (59)    div (60)    dosxterr (61)    dostounix (62)  
drawpoly (63)    dup (64)    dup2 (65)

E

ecvt (67)    ellipse (68)    \_\_emit\_\_ (69)    enable (69)  
eof (70)    exec... (71)    \_exit (75)    exit (75)    exp (76)

F

fabs (77)    farcalloc (78)    farcoreleft (78)    farfree (79)

farheapcheck (80)    farheapcheckfree (80)    farheapchecknode (82)  
 farheapfillfree (83)    farheapwalk (84)    farmalloc (85)  
 farrealloc (85)    fclose (86)    fcloseall (87)    fcvt (87)  
 fdopen (88)    feof (89)    ferror (89)    fflush (90)    fgetc (91)  
 fgetchar (92)    fgetpos (92)    fgets (93)    filelength (94)  
 fileno (94)    fillellipse (96)    fillpoly (97)    findfirst (98)  
 findnext (98)    floodfill (99)    floor (100)    flushall (101)  
 \_fmemccpy (101)    \_fmemchr (101)    \_fmemcmp (102)    \_fmemcpy (102)  
 \_fmemicmp (103)    \_fmemset (103)    fmod (103)    fnmerge (104)  
 fnsplit (105)    fopen (106)    FP\_OFF (107)    \_fpreset (108)  
 fprintf (109)    FP\_SEG (109)    fputc (110)    fputchar (110)  
 fputs (111)    fread (111)    free (112)    freemem (113)    freopen (113)  
 frexp (114)    fscanf (114)    fseek (115)    fsetpos (116)  
 fstat (117)    \_fstrcat (118)    \_fstrchr (118)    \_fstrcspn (119)  
 \_fstrdup (119)    \_fstricmp (119)    \_fstrlen (120)    \_fstrlwr (120)  
 \_fstrncat (121)    \_fstrncmp (121)    \_fstrncpy (122)    \_fstrnicmp (122)  
 \_fstrnset (123)    \_fstrprbk (123)    \_fstrrchr (123)    \_fstrrev (124)  
 \_fstrset (124)    \_fstrspn (125)    \_fstrstr (125)    \_fstrtok (125)  
 \_fstrupr (126)    ftell (126)    ftime (126)    fwrite (127)

## G

gcvt (127)    geninterrupt (128)    getarccoords (129)  
 getasptratio (130)    getbkcolor (131)    getc (133)    getcbrk (133)  
 getch (134)    getchar (134)    getche (135)    getcolor (135)  
 getcurdir (136)    getcwd (137)    getdate (138)  
 getdefaultpalette (139)    getdfree (140)    getdisk (141)  
 getdrivename (141)    getdta (142)    getenv (143)    getfat (144)  
 getfatd (145)    getfillpattern (146)    getfillsettings (146)  
 getftime (148)    getgraphmode (150)    getimage (151)  
 getlinesettings (153)    getmaxcolor (155)    getmaxmode (156)  
 getmaxx (157)    getmaxy (158)    getmodename (159)    getmoderange (160)  
 getpalette (161)    getpalettesize (162)    getpass (163)    getpid (163)  
 getpixel (165)    getpsp (166)    gets (166)    gettext (167)  
 gettextinfo (168)    gettextsettings (169)    gettime (171)  
 getvect (171)    getverify (172)    getviewsettings (173)    getw (174)  
 getx (175)    gety (176)    gmtime (177)    gotoxy (178)  
 graphdefaults (179)    grapherrormsg (180)    \_graphfreemem (181)  
 \_graphgetmem (182)    graphresult (184)

## H

harderr (186)    hardresume (189)    hardretn (191)    heapcheck (192)  
 heapcheckfree (193)    heapchecknode (194)    heapfillfree (196)  
 heapwalk (197)    highvideo (198)    hypot (198)

## I

imag (199)    imagesize (199)    initgraph (201)    inport (204)  
 inportb (205)    inline (206)    installuserdriver (206)  
 installuserfont (208)    int86 (209)    int86x (210)    intdos (211)  
 intdosx (212)    intr (213)    ioctl (214)    isalnum (215)  
 isalpha (215)    isascii (215)    isatty (216)    iscntrl (216)  
 isdigit (216)    isgraph (217)    islower (217)    isprint (217)    ispunct (217)    isspa  
 ce (218)    isupper (218)    isxdigit (218)  
 itoa (218)

## K

kbhit (219)      keep (219)

## L

labs (221)      ldexp (221)      ldiv (221)      lfind (222)      line (222)  
linerel (223)      lineto (225)      localeconv (226)      localtime (226)  
lock (227)      log (228)      log10 (228)      longjmp (230)      lowvideo (230)  
\_lrotl (231)      \_lrotr (231)      lsearch (231)      lseek (232)      ltoa (233)

## M

malloc (234)      matherr (235)      max (235)      memccpy, \_fmemccpy (236)  
memchr, \_fmemchr (237)      memcmp, \_fmemcmp (238)      memcpy, \_fmemcpy (238)  
memicmp, \_fmemicmp (239)      memmove (239)      memset, \_fmemset (239)  
min (240)      mkdir (240)      MK\_FP (241)      mktemp (241)      mktime (242)  
modf (243)      movedata (243)      moverel (244)      movetext (245)  
moveto (246)      movmem (247)

## N

norm (247)      normvideo (248)      nosound (248)

## O

\_open (248)      open (249)      outport (251)      outportb (251)  
outtext (251)      outtextxy (253)      \_OvrInitEms (254)      \_OvrInitExt (254)

## P

parsfnm (255)      peek (256)      peekb (257)      perror (258)      pieslice (259)  
poke (260)      pokeb (260)      polar (261)      poly (261)      pow (262)  
pow10 (262)      ...printf (262)      putc (269)      putchar (270)      putchar (270)  
putenv (271)      putimage (272)      putpixel (274)      puts (275)  
puttext (276)      putw (276)

## Q

qsort (277)

## R

raise (278)      rand (279)      randbrd (279)      randbwr (281)      random (282)  
randomize (283)      \_read (283)      read (284)      real (285)      realloc (286)  
rectangle (286)      registerbgidriver (287)      registerbgifont (289)  
remove (290)      rename (290)      restorecrtmode (291)      rewind (292)  
rmdir (293)      \_rotl (294)      \_rotr (294)

## S

sbrk (295)      scanf (295)      searchpath (302)      sector (303)  
segread (304)      setactivepage (304)      setallpalette (306)  
setaspectratio (308)      setbkcolor (309)      setblock (311)      setbuf (312)  
setcbrk (313)      setcolor (314)      \_setcursortype (316)      setdate (316)  
setdisk (317)      setdta (318)      setfillpattern (319)  
setfillstyle (320)      setftime (322)      setgraphbufsize (323)  
setgraphmode (324)      setjmp (325)      setlinestyle (326)  
setlocale (328)      setmem (329)      setmode (329)      setpalette (330)  
setrgbpalette (332)      setttextjustify (333)      setttextstyle (335)  
settime (337)      setusercharsize (338)      setvbuf (339)      setvect (340)  
setverify (341)      setviewport (341)      setvisualpage (342)  
setwritemode (344)      signal (345)      sin (346)      sinh (346)  
sleep (347)      sopen (347)      sound (349)      spawn... (350)  
sprintf (352)      sqrt (352)      srand (353)      sscanf (353)      stat (354)

\_status87 (355) stime (356) stpcpy (356) strcat, \_fstrcat (356)  
strchr, \_fstrchr (361) strcmp (361) strcmpi (361) strcoll (362)  
strcpy (363) strcspn, \_fstrcspn (363) strdup, \_fstrdup (364)  
\_strerror (364) strerror (365) strptime (365)  
stricmp, \_fstricmp (367) strlen, \_fstrlen (367) strlwr, \_fstrlwr (367)  
strncat, \_fstrncat (368) strncmp, \_fstrncmp (368) strncmpi (369)  
strncpy, \_fstrncpy (370) strnicmp, \_fstrnicmp (370) strnset, \_strnset (371)  
strpbrk, \_fstrpbrk (371) strrchr, \_fstrchr (372) strrev, \_fstrrev (372)  
strset, \_fstrset (372) strspn, \_fstrspn (373) strstr, \_fstrstr (373)  
strtod (374) strtok, \_fstrtok (374) strtol (374) strtoul (375)  
strupr, \_fstrupr (375) strxfrm (376) swab (376) system (377)

## T

tan (377) tanh (378) tell (378) textattr (379)  
textbackground (380) textcolor (381) textheight (382)  
textmode (383) textwidth (384) time (385) tmpfile (386)  
tmpnam (386) toascii (387) \_tolower (387) tolower (387)  
\_toupper (388) toupper (388) tzset (389)

## U

ultoa (389) ungetc (389) ungetch (390) unixtodos (391)  
unlink (391) unlock (392)

## V

va \_arg, va \_end, va \_start (393) vfprintf (395) vprintf (395)  
vscanf (396) vsprintf (397) vsscanf (397)

## W

wherex (398) wherey (399) window (399) \_write (401) write (401)

## 第三章 全局变量 (402)

\_8087 (402) \_argc (402) \_argv (402) \_ctype (402)  
daylight (402) directvideo (403) environ (403)  
errno, \_doserrno, sys\_errlist, sys\_nerr (403)  
\_fmode (405) \_heaplen (406) \_openfd (406)  
\_osmajor, \_osminor (406) \_psp (406) \_stklen (407)  
timezone (407) tzname (407) \_version (408) \_wscroll (408)



# 简介

本书包含所有 Borland C++ 库例程、公用变量、公用定义类型的定义，以及用于说明如何使用这些例程、变量和类型的例子程序代码。

如果你是 C 或 C++ 程序设计语言的初学者，或者要查找有关 Borland C++ 手册的内容，请阅读《入门》的简介和第二章“Borland C++ 手册导航”。

以下是本书的概述：

第一章：“main 函数”讨论了 main 函数的参数（包括通配符参数），给出了一些例子程序，并给出了关于 Pascal 调用和 main 返回值的约定。

第二章：“运行库”按字母顺序列出了所有 Borland C++ 库函数。每一个函数都给出了用法，包括文件、操作说明、相关函数、返回值、函数的移植性信息及如何使用该函数的例子。

第三章：“全局变量”定义和讨论了 Borland C++ 的全局变量。使用这些变量可以节省大量的程序设计时间。

## 类和成员函数文档

第二章中包括了一些类和成员函数，以下是类和成员函数的清单：

| 名       | 类型   |
|---------|------|
| abs     | 成员函数 |
| acos    | 成员函数 |
| arg     | 成员函数 |
| asin    | 成员函数 |
| atan    | 成员函数 |
| bcd     | 类    |
| complex | 类    |
| conj    | 成员函数 |
| cos     | 成员函数 |
| cosh    | 成员函数 |
| exp     | 成员函数 |

|       |      |
|-------|------|
| imag  | 成员函数 |
| log   | 成员函数 |
| log10 | 成员函数 |
| norm  | 成员函数 |
| pow   | 成员函数 |
| polar | 成员函数 |
| real  | 成员函数 |
| sin   | 成员函数 |
| sinh  | 成员函数 |
| sqrt  | 成员函数 |
| tan   | 成员函数 |
| tanh  | 成员函数 |

# 第一章 main函数

每个C和C++程序都必须有一个main函数，有些编程人员将main放在文件的开头，有些程序员将它放在文件尾。不管它的位置在哪里，都必须遵循以下要点：

## main的变元

Borland C++的启动程序将argc、argv和env三个参数(变元)传给main。

■ argc是传给main的命令行参数的个数，它是一个整型数。

■ argv是一个指向字符串的指针的数组。

- 在DOS 3.0或以上版本中，argv (0)是运行的程序的全名；
- 在DOS 3.0以下版本中，argv (0)指向空串；
- argv (1)指向在DOS命令行上紧接运行程序名后敲入的第一个字符串；
- argv (2)指向在运行程序名后敲入的第二个字符串；
- argv (argc-1)指向传给main的最后一个参数；
- argv (argc)为空

■ env也是一个指向字符串的指针的数组。每个env ()元素都是ENVVAR=value形式的字符串。

· ENVVAR是环境变量名，比如PATH；

· value是ENVVAR设置的值，比如C:\DOS;C:\TOOL；。

如果声明了这些变量，就必须严格按照argc，argv，env的顺序声明它们。例如，以下都是有效的main参数声明：

```
main ()
main (int argc)
main (int argc, char *argv)
main (int argc, char *argv, char *env)
```

声明main (int argc)是合法的，但一般不单独使用argc，而将它与argv一起使用。

参数env也可通过全局变量environ获得。参阅第三章的environ和第二章的getenv和putenv项。

argc和argv也可通过全局变量\_argc和\_argv获得。

## 例子

下面这个例子程序ARGS.EXE说明了如何使用传给main的参数：

```
/* Program ARGS.C */
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv (), char *env ())
{
    int;

    printf ("The value of argc is %d \n \n ", argc);
    printf ("These are the %d command-line arguments passed to
            main: \n \n ", argc);
    for (i = 0; i < argc; i++)
        printf (" argv (%d): %s\n ", i, argv (i));

    printf ("\nThe environment string (s) on this system are: \n \n ");
    for (i = 0; env (i) != NULL; i++)
        printf (" env (%d): %s\n ", i, env (i));

    return 0;
}
```

假设在DOS提示符以如下命令行来运行ARGS.EXE：

```
c:>args first_arg "arg with blanks" 3 4 "last but one" stop!
```

注意：可以用双引号将空格括起来，如本例所示。

ARGS.EXE的输出为：

```
The value of argc is 7
```

```
These are the 7 command-line arguments passed to main:
```

```
argv (0): C:\BC\TESTARGS.EXE
argv (1): first_arg
argv (2): arg with blanks
argv (3): 3
argv (4): 4
argv (5): last but one
argv (6): stop!
```

```
The environment string (s) on this system are:
```

```
env (0): COMSPEC=C:\COMMAND.COM
env (1): PROMPT=$p $g
env (2): PATH=C:\SPRINT;C:\DOS;C:\BC
```

可传给main的最大命令行长度为128个字符(包括其中的空格)，这是DOS的限制。

## 通配符参数

包括通配符的命令行参数可扩展为所有匹配的文件名，就象DOS的COPY命令一样。所要做的是将程序与WILDARGS.OBJ目标文件相连接。

将WILDARGS.OBJ连入程序代码后，就可以向main函数传递\*.\*等通配符。参数将在argv数组中扩展为所有匹配的文件名。argv数组的最大长度取决于可用的堆内容。

如果没有找到匹配的文件，就不改变传递的参数(即向main传递包含通配符的字符串)。

## 使用-P选项(Pascal调用约定)

如果以Pascal调用约定编译程序，务必显式地将main声明为C类型。这可用cdecl来实现，如下所示：

```
cdecl main (int argc, char *argv (), char *envp ())
```

main的返回值

main返回的是程序的状态码，这是一个整数。但如果程序通过调用例程exit(或\_exit)来退出，则main的返回值为传给exit或\_exit的值。

如果在IDE中运行程序，可以选择File | Get Info来了解返回值。

## 第二章 运行库

本章详细讨论Borland C++库中的每个函数。少量例程由于招待相似或相关的任务而被按“族”组合到一起(如exec...和spawn...)。其余例程都单独列为一项。

### 函数项说明

|      |  |
|------|--|
| 函数名  | 功能   |
| 用法   | <code>#include &lt;header.h&gt;</code><br>本部分列出包含函数原型或定义函数用到的常量、枚举类型的头文件。<br><code>function (modifier parameter (, ...));</code><br>本部分给出函数的用法、初始化的参数名 |
| 原型在  | <code>header.h</code><br>这部分列出包含函数原型的头文件。有些函数的原型包含在多个头文件中，此时，就列出这些文件。  |
| 说明   | 本部分说明函数的作用、采用的参数及使用函数时的细节。   |
| 返回值  | 函数的返回值。  |
| 可移植性 | 列出该函数适用的系统和语言。包括UNIX、IBM PC及C++、Windows、ANSI C   |
| 兼容性  | 的兼容性。  |
| 参 见  | 与本函数相关的例程。   |
| 例 子  | 本部分给出说明如何使用该函数的例子。   |

abort 异常终止一进程

用法 #include <stdlib.h>  
void abort(void);

原型在 stdlib.h, process.h

说明 本函数向stderr写终止信息"Abnormal program termination", 并通过调用带终止码3的\_exit来终止程序。

返回值 将终止码3返回给父进程或DOS。

可移植性 适用于UNIX系统, 在ANSI C中定义。

参 见 assert, atexit, exit, \_exit, raise, signal, spawn系统函数。

例 子

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(void)
{
    printf("Calling abort()\n");
    abort();
    return 0; /* This is never reached */
}
```

abs 绝对值

用法 实型版本: #include <math.h>      复数版本: #include <complex.h>

int abs(int i);      int abs(complex i);

原型在 实型版本: stdlib.h, math.h      复数版本: stdlib.h, complex.h

说明 本函数返回整数参数i的绝对值, 如果包含了stdlib.h头文件并调用了abs, abs将作为可扩展为插入代码的宏看待。如果不包含stdlib.h (或包含了但使用了#undef abs), abs将作为函数看待而不是宏。

cabs是一计算复数znum绝对值的宏。znum是一具有类型complex的结构。

此结构在math.h定义如下:

```
struct complex {
    double x,y;
};
```

调用cabs等价于调用具有znum的实部和虚部的sqrt, 如下式所示:

```
sqrt(znum.x*znum.x+znum.y*znum.y)
```

如果没有包含math.h头文件(或包含了而又使用了#undef cabs), cabs将作为一函数看待而不是宏。

fabs计算double类型x的绝对值。

labs计算long整型n的绝对值。

返回值 abs返回0到32767间的一个整数, 有一个例外: 当参数值为-32768时, 返回-32768。

cabs返回double型的znum的绝对值, 如果发生溢出, 将返回HUGE\_VAL并置error为:

ERANG      结果超出范围

可以通过matherr 函数修改对cabs的错误处理。

fabs返回x 的绝对值, lass 返回n 的绝对值, 它们都不返回出错信息。

可移植性 适用于UNIX 系统

参 见 matherr

例 子

```
#include <stdio.h>
#include <math.h>
```

```
int main (void)
{
    int number = -1234;

    printf ("number: %d  absolute value: %d \n ", number, abs(number));
    return 0;
}
```

absread 读数据

用 法 int absread(int drive,int nsects,int sectno,  
void \*buffer);

相关函数

用 法 int abswrite (int drive,int nsects,int sectno,  
void \*buffer);

原 型 在 dos.h

说 明 这些函数读和写磁盘的指定扇区内容。它们忽略磁盘的逻辑结构, 并不关心文件、FAT 和目录。

absread通过DOS 中断0x25 读指定的磁盘扇区内容。

abswrite通过DOS 中断0x26写内容到指定的磁盘扇区。

drive =要读的磁盘驱动器号(0=A,1=B 等)

nsects =要读的扇区数

sectno =要读的起始扇区号

buffer =数据读或写的存储区地址

读的扇区数受段中buffer 以上的存储空间量的限制。因此, 在一次调用absread 或abswrite 过程中, 最大的可读存储空间为64k 字节。

返 回 值 如果调用成功, 两函数都返回0。

如果调用出错, 都返回-1, 并置error 为经系统调用后返回AX 寄存器的值。

关于error 的说明, 请参阅MS -DOS 使用手册。

可移植性 只适用于MS -DOS

例 子

```
/* absread example */
```

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dos.h>
```

```
int main (void)
{
    int i, strt, ch_out, sector;
```



```

char buf(512);

printf ("Insert a diskette into drive A and press any key \n ");
getch ();
sector = 0;
if (absread(0, 1, sector, &buf) != 0)
{
    perror ("Disk problem ");
    exit (1);
}
printf ("Read OK\n ");
strt = 3;
for (i=0; i<80; i++)
{
    ch_out = buf(strt+i);
    putchar(ch_out);
}
printf ("\n ");
return (0);
}

```

**abswrite** 写数据

用法 `int abswrite (int drive, int nsects, int sectno, void *buffer);`

原型在 `dos.h`

说明 见 `absread`

**access** 确定文件的存取权限

用法 `int access (char *filename, int amode);`

原型在 `io.h`

说明 本函数检查给定名的函数是否存在，它的可读、可写和可执行性。`filename` 是指向文件名字符串的指针。

包含在 `amode` 中的模式结构如下：

- 06 检查读/写允许
- 04 检查读允许
- 02 检查写允许
- 01 执行(被忽略)
- 00 检查文件的存在性

注意：在 MS-DOS 下，所有存在的文件具有读访问 (`amode=04`) 性，所以 00 和 04 有同样的结果。同理，06 和 02 是等价的，因为在 MS-DOS 下，写访问性隐含了读访问性。

如果 `filename` 指向一目录，`access` 就只确定该目录是否存在。

返回值 如果要求的存取是允许的，返回 0；否则返回 -1，并置 `error` 为以下值之一：

- `ENOENT` 路径或文件名没找到
- `EACCES` 无此权限

可移植性 适用于 UNIX 系统

参见 `chmod`

例子

```
#include <stdio.h>
```