萧 柔 潘金贵 吴卫华等 编著
陈世福 顾铁成 审校

Borland C++
实用库函数大全

南京大学出版社

(苏) 新登字第 011 号

内 容 简 介

C 和 C++语言的功能与它所提供的库函数直接相关。Borland C++提供了 500 多个库函数，使用这些函数可以完成诸如输入输出、数学计算、图形设计、进程控制等广泛的任务，并可提高程序开发的速度和质量。

本书系统地介绍了面向对象的程序设计语言 Borland C++的所有运行库函数的完整信息，详细讲解了每个库函数、公用变量、公用定义类型，还给出了说明如何使用这些函数、变量和类型的例子。

本书与其姊妹篇 —— 《Borland C++程序设计教程》和《Borland C++程序开发工具》是一套学习和使用 Borland C++不可缺少的工具书和参考书。既可作为高等院校计算机专业的本科生和研究生学习 C 语言和 C++语言的教材或教学参考书，亦可供有关专业的学生以及从事计算机系统软件及应用软件开发工作的工程技术人员作为 Borland C++、Turbo C++软件包的使用手册和参考书使用。

欲了解与本书配套的例子盘和系统盘的信息可与编者联系。

Borland C++实用库函数大全

萧柔 潘金贵 吴卫华等 编著

陈世福 顾铁成 审校

南京大学出版社出版

(南京大学校内)

江苏省新华书店发行 江苏射阳印刷厂印刷

开本：787×1092毫米 1/16 印张：34 字数：840千字

1992年10月第1版 1994年4月第2次印刷

印数 6001－12000

ISBN 7－305－01662－4／TP·49

定价：21.00元

责任编辑：顾其兵

前　言

一、面向对象的程序设计与 C++

面向对象程序设计方法提出了一种全新的程序设计思想，把数据和对数据的操作封装起来。目前，对面向对象的软件设计技术的研究和应用在我国正蓬勃发展，并越来越流行。

C++是一种面向对象的程序设计语言，它是 C 语言的进一步发展。C++语言是可移植的，可以很容易地将一个用 C++语言编写的程序从一个计算机系统移植到另一个计算机系统上。因此，可以毫不夸张地说，任何应用程序或系统程序均可用 C++语言来开发。

二、Borland C++

1. 语言特色

美国 Borland 国际公司在 Turbo C 的基础上，于 1990 年推出过面向对象的程序设计语言 —— Turbo C++，它继承并发展了原来 Turbo C 集成开发环境的优良特性，并引入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的程序设计工具。

1990 年 5 月，美国 Microsoft 公司推出了 Windows 3.0，立刻风靡全球，在微机软件界引起了一场轰动。国际上许多软件公司为适应这场变革，纷纷推出了适用于 Windows 3.0 的软件包，或者对原有的流行软件进行改造，推出了适用于 Windows 的版本。Borland 国际公司为适应这一潮流，于 1991 年 2 月推出了适用于开发 Windows 应用软件的 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 还具有以下新特色：

· 与 AT&T C++和 ANSI C 完全兼容：Borland C++提供了用 C++方法进行编程的全部功能，完整地实现了 AT&T C++ 2.0 版的功能。

· Borland C++也实现了迄今为止最新的 ANSI C 标准。并且完全支持 Kernigham 和 Ritchie 的定义。另外，Borland C++包含有一定的扩展，使得程序员能够用混合语言和混合模型编程来扩充 PC 机的能力。

用户可以设置编译选项，以便在碰到扩充的部分时可以得到提示或警告。也可以设置编译选项，使编译器同等对待 Borland C++中扩充的关键字和常规标识符。

Borland C++中还扩充了 ANSI C 中#pragma 指令所具有的处理非标准的与具体实

现细节无关的功能。

借助 Borland C++，用户可以调用 ANSI C 及 C++的全部功能。可以单独用 ANSI C 编程，也可以将 C 程序转换成 C++程序。

为了帮助用户入门，Borland C++中提供了一些 C++类库。这些类库使用类来完成各种功能。并提供了对 C++ 2.0 版流的支持。

· 支持 Microsoft Windows 环境下的应用程序开发：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。例子盘中提供了一些用 C++和 C 编制的 Windows 应用程序的实例，有助于读者理解这方面的内容。

· 提供了预编译的头文件，利用它可以缩短程序的编译时间。

· 提供了实模式和保护模式的编译器。Borland C++包括四种编译器，它们分别为实模式和保护模式下的程序员平台和命令行编译器。每类编译器都可以编译 C 和 C++程序。在保护方式下运行 C++可以取得非常高的效率，因为它不存在内外存交换的问题。

· 类库中提供了集合、数组等类型。

· 求助系统中添加了 Windows API（应用程序接口）。

· 崭新的 Borland 程序员平台：它的新一代的用户接口，其性能大大优于旧式的集成开发环境，它提供了对计算机上全范围内程序和工具的访问能力。无论是在实模式还是在保护模式下运行，它都能支持以下功能：

鼠标器；

多重覆盖窗口；

多重文件编辑；

内部汇编码；

嵌入式汇编器；

集成调试器；

对大缓冲区的恢复（undo）和重复（redo）功能、等等。

· VROOMM 提供面向对象的实时虚拟存储管理：VROOMM 使程序员能够很简单地覆盖代码。程序员只须为覆盖选择代码段，其他均由 VROOMM 自动处理，可使用户的代码保存在 640K 的空间内。

· 新程序员平台的联机访问。

· 联机的快速文本求助，用户可用拷贝的模拟程序例子试验每一个函数。

· 有着许多独立的库函数，包括堆检测函数和完整的复数，以及 BCD 数学函数集，能快速完成复杂的算术运算和复数运算。

其他特征包括：

· 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

· 候补配置文件：用户可以创建几个配置文件，并可在需要的时刻使用其中合适的一个。

· 用于命令行编译的响应文件。

· 快速完成复杂的算术运算和复数运算。

2. 硬件和软件环境

Borland C++运行于 IBM PC 系列计算机上，包括 XT、AT、PS／2 和 IBM 全兼容机。Borland C++要求机器至少配有 3.0 以上版本的 DOS 和至少的 640K 内存。监视器为 80 列即可。最低的要求是一个硬盘和一个软盘驱动器。

Borland C++中包含了利用 8087 协处理器进行浮点运算的例程。如果没有协处理器，它也可以进行仿真运算。虽然运行 Borland C++时不必有 8087 协处理器，但 8087 协处理器可以有效地提高程序的运行效率。

Borland C++可支持鼠标器。如果配有鼠标器，必须与下列要求之一全兼容。

- Microsoft Mouse 6.1 版或更高版本，或者与此兼容的 mouse 版本。
- Genus Mouse 9.0 版或更高版本。
- IMSI Mouse 6.11 版或更高版本。
- Logitech Mouse 3.4 版或更高版本。
- Mouse Systems/PC Mouse 6.22 版或更高版本。

三、本书内容安排

为了使我国广大的计算机工作人员能够更好地利用 Borland C++进行程序设计工作，在南京大学出版社的支持下，我们根据长期从事 C 语言编程和面向对象程序设计的经验，参阅了国内外的有关软件、资料，编译这套 Borland C++程序设计丛书，第一批包括《Borland C++程序设计教程》、《Borland C++程序开发工具》和《Borland C++实用库函数大全》。1993 年还将推出第二批。它们全面、系统地介绍了 Borland C++的基础知识和高级程序设计技术，是一套引导读者进行 C、C++面向对象程序设计和进行 Windows 应用编程的参考书。

本书系统地介绍了 Borland C++的所有库函数、公用变量、公用定义类型的定义，还给出了说明如何使用这些函数、变量和类型的例子，并提供了查找这些库函数和公用变量的索引。

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，并引出了关于 Pascal 调用和 main 返回值的说明。

第二章"**Borland C++运行库函数分类**"简介了十六类库函数以及类和成员函数。

第三章"**Borland C++运行库函数详解**"按字母顺序列出了 Borland C++的所有库函数。对每一个函数都给出了作用、用法、包含头文件、使用说明、相关函数、返回值、函数的可移植性等信息，以及如何使用该函数的例子。

第四章"**全局变量**"定义和讨论了 Borland C++的全局变量。使用这些变量可以大大节省编程时间。

附录一"**库函数索引**"给出了查找第三章详细介绍的 482 个库函数说明页的索引，利用它能快速地找到有关的信息。

附录二"**全局变量索引**"给出了查找 Borland C++使用的 20 个全局变量说明页的索引。

Borland C++适用于需要功能强大、快速和高效编译器的 C++和 C 程序员，他们可以用它来创建应用程序，包括 Microsoft Windows 应用程序；适用于所有想利用 Turbo 优点而学习 C++或 C 语言的 Turbo Pascal 程序员；适用于正在学习 C++或 C 语言的读者；Borland C++还适用于使用 AT&T 公司的 C++ 2.0 和 ANSI C 的程序员。

对于 C 或 C++程序设计语言的初学者，我们建议读者首先阅读由南京大学出版社出版的《学习和使用 Turbo C 语言》（上、下册）和本书的姊妹篇《Borland C++程序设计教程》和《Borland C++程序开发工具》。

参加本书编著工作的人员有潘金贵、萧柔、吴卫华、顾铁成、虞育新、潘东旭、冀惠刚、忻超、李竹华、王勇、金子方等，潘金贵和萧柔同志对全书进行了仔细的修改和统编。

本书承蒙南京大学计算机科学系陈世福教授主审，顾铁成老师协助校阅了有关章节，在此谨向他们深表谢意。

由于水平、时间所限，不妥之处在所难免，欢迎广大读者批评指正。

编著者

一九九二年三月于南京

目 录

第一章 main 函数

每个 C 和 C++程序都必须有一个 main 函数，有些程序员将 main 放在源程序的开头，有些程序员将它放在源程序的末尾。不管 main 函数的位置在哪里，都必须遵循以下规定。

1.1 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 函数。

(1) argc 是传给 main 的命令行参数的个数，它是一个整型数。

(2) argv 是一个指向字符串的指针数组。

- 在 DOS 3.0 或以上版本中，argv[0]是运行程序的全名；
- 在 DOS 3.0 以下版本中，argv[0]指向一个空串；
- argv[1]指向在 DOS 命令行上紧接运行程序名后键入的第一个字符串；
- argv[2]指向在运行程序名后键入的第二个字符串；
- argv[argc-1]指向传给 main 的最后一个参数；
- argv[argc]为空。

(3) env 也是一个指向字符串的指针数组。每个 env[ ]元素都是"ENVVAR=value"形式的字符串。

- ENVVAR 是环境变量名，例如 PATH。
- value 是 ENVVAR 设置的值，例如 C:\DOS；C:\TOOL。

如果要定义这些变量，就必须严格按照 argc、argv、env 的顺序定义它们。例如，以下都是有效的 main 参数定义：

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

以下这个定义也是合法的，但一般不单独使用 argc，而是将它与 argv 一起使用。

```
main(int argc)
```

也可以通过全局变量 environ 获得参数 env。请读者参阅本书第三章的 environ 及第二章的 getenv 和 putenv 项。

还可以通过全局变量__argc 和__argv 访问 argc 和 argv。

1.2 例　子

下面的例子程序说明了如何使用传给 main 的参数:

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

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

    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 命令行提示符下通过如下命令①来运行 ARGS.EXE:

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

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:
```

①**注意**: 可以用双引号将空格括起来，如本例所示。

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

传给 main 的命令行的最大长度为 128 个字符（包括其中的空格），这是 DOS 系统的限制。

1.3 通配符参数

可以将包括通配符（* 和?）的命令行参数扩展为所有匹配的文件名，就像 DOS 的 COPY 命令一样。程序员所要做的工作是将程序与 WILDARGS.OBJ 目标文件相连接。

将 WILDARGS.OBJ 连入程序代码后，就可以向 main 函数传递诸如"*.*"这样的通配符参数，参数将会在 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大小。

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）。

1.4 使用-P 选项（Pascal 调用约定）

如果以 Pascal 调用约定编译程序，就必须显式地将 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[ ])
```

1.5 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。

例如，程序中使用了调用语句：

```
exit(1)
```

则返回值为 1。

如果在集成环境（IDE）中运行程序，可以选择 File|Get Info 来了解返回值。

第二章　Borland C++库函数分类

Borland C++提供了482个函数和宏，可以在C和C++程序中调用它们来执行各种任务，包括低级和高级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。

Borland C++的例程被放在库文件中（Cx.LIB，CPx.LIB，MATHx.LIB和GRAPHICS.LIB）。因为Borland C++支持六种不同的内存模式，除了tiny模式之外，每种模式都有自己的库文件和数学文件，其中包含对应该特定模式的例程版本。（tiny模式与small模式共享库文件和数学文件。）

Borland C++实现的是ANSI C标准，它允许向C程序中例程提供函数原型。所有Borland C++的库例程都在一个或多个文件中用原型进行声明。在每个库函数后的圆括号内，我们给出了该函数的原型说明文件。

本章我们先列出并说明头文件，然后汇总库例程执行的各种任务，即每个库函数所属的功能类别。

2.1　Borland C++头文件

头文件也叫嵌入文件或包含文件，它提供了库函数的函数原型说明，库函数所用的数据类型和符号常量也在这些文件中定义。另外它还定义了Borland C++库函数中使用的全局变量。Borland C++库在头文件名及其常量名方面遵循ANSIC标准。

表2.1　Borland C++头文件一览表

头文件名	作　用
alloc.h	说明内存管理函数（分配、回收等）
assert.h*	定义assert调试宏
bcd.h	说明C++类bcd以及bcd重载操作符和bcd数学函数
bios.h	说明用于IBM PC的ROM BIOS例程调用中的各种函数
complex.h	说明C++数学复数函数
conio.h	说明用于DOS控制台I/O例程调用的各种函数
ctype.h*	含有字符分类和字符转换宏（如isalpha和toascii）
dir.h	含有用于目录和路径名的结构、宏和函数
dos.h	定义各种常量并给出DOS和8086调用所需的说明
errno.h*	定义错误代码的常量助记符
fcntl.h	定义用于与库例程open连接的符号常量
float.h*	含有浮点例程的参数
fstream.h	说明支持文件输入和输出的C++流类
generic.h	包含通用类说明的宏

表 2.1 (续)

头文件名	作　用
graphics.h	说明图形函数原型
io.h	含有低级输入／输出例程的结构和说明
iomanip.h	说明 C++流 I／O 操纵符，并含有建立参数化操纵符的宏
iostream.h	说明基本 C++流 I／O 例程
limits.h*	含有环境参数、编译时刻限制信息和整型量范围
locale.h*	说明提供特定国家和语言信息的函数
math.h*	说明数学函数的原型;定义宏 HUGE__VAL;说明 matherr 例程所用的异常结构
mem.h	说明内存操作函数（其中有许多还定义在 string.h 中）
process.h	含有 spawn...和 exec...函数的结构和说明
setjmp.h*	定义 longjmp 和 setjmp 函数所用的类型 jmp__buf,并说明例程 longjmp 和 setjmp
share.h	定义用在利用文件共享的函数中的参数
signal.h*	定义 signal 和 raise 函数所需要的常量和说明
stdarg.h*	定义用于读取说明为接收一个变数目变量的函数的变元表的宏
stddef.h*	定义几个公共数据类型和宏
stdio.h*	定义 UNIX System V 下在 Kernighan 和 Ritchie 中定义的标准 I／O 软件包及其扩展所需的类型和宏。定义标准 I／O 预定义流 stdin、stdout、stdprn 和 stderr，并说明流级 I／O 例程
stdiostr.h	说明 stdio FILE 结构所用的 C++（2.0 版本）流类
stdlib.h*	说明几个常用的例程：转换例程、搜索／排序例程以及别的例程
stream.h	说明 C++（1.2 版本）流（I／O）例程
string.h*	说明几个串操作和内存操作例程
strstrea.h	说明内存中字节数组所用的 C++流类
sys\stat.h	定义用于打开和建立文件的符号常量
sys\timeb.h	说明函数 ftime 和 ftime 返回的结构 timeb
sys\types.h	说明时间函数所用的类型 time__t
time.h*	定义一个由时间转换例程 asctime、localtime 和 gmtime 填充的结构；定义例程 ctime、difftime、gmtime、localtime 和 stime 所用的类型；提供这些例程的原型
values.h	定义重要常量，其中包括机器独立性。这是为与 UNIX System V 兼容而提供的

带星号(*)的项表示与 ANSI C 兼容。

2.2 各类库例程简介

Borland C++的库例程完成各种任务。本节中，我们列出执行各种任务的例程，以及说明它们的包含文件。有关这些函数的完整信息，参见第三章中对每个库函数项的详细介绍。

2.2.1 分类例程

以下例程把 ASCII 字符分为字母、控制字符、标点符号、大写字体等。

isalnum	(ctype.h)	iscntrl	(ctype.h)
isalpha	(ctype.h)	isdigit	(ctype.h)
isascii	(ctype.h)		

isgraph	(ctype.h)	issapace	(ctype.h)
islower	(ctype.h)	isupper	(ctype.h)
isprint	(ctype.h)	isxdigit	(ctype.h)
ispunct	(ctype.h)		

2.2.2 转换例程

以下这些例程把字符和串从字母转换为各种数值表示（浮点数、整型数和长整型），以及相反方向的转换，还有大写与小写之间的转换。

atof	(stdlib.h)	strtol	(stdlib.h)
atoi	(stdlib.h)	strtoul	(stdlib.h)
atol	(stdilib.h)	toascii	(ctype.h)
ecvt	(stdlib.h)	tolower	(ctype.h)
fcvt	(stdlib.h)	toupper	(ctype.h)
gcvt	(stdlib.h)	ultoa	(stdlib.h)
itoa	(stdlib.h)	_tolower	(ctype.h)
ltoa	(stdlib.h)	_toupper	(ctype.h)
strtod	(stdlib.h)		

2.2.3 目录控制例程

以下这些例程对目录和路径名进行操作。

chdir	(dirh.)	getdisk	(dir.h)
findfirst	(dir.h)	mkdir	(dir.h)
findnext	(dir.h)	mktemp	(dir.h)
fnmerge	(dir.h)	rmdir	(dir.h)
fnsplit	(dir.h)	searchpath	(dir.h)
getcurdir	(dir.h)	setdisk	(dir.h)
getcwd	(dir.h)		

2.2.4 诊断例程

以下这些诊断例程提供内部故障诊断功能。

assert	(assert.h)	perror	(errno.h)
matherr	(math.h)		

2.2.5 图形例程

以下这些例程使程序能够使用图形功能。

arc	(graphics.h)	cleardevice	(graphics.h)
bar	(graphics.h)	clearviewport	(graphics.h)
bar3d	(graphics.h)	closegraph	(graphics.h)
circle	(graphics.h)	detectgraph	(graphics.h)

drawpoly	(graphics.h)
ellipse	(graphics.h)
fillellipse	(graphics.h)
fillpoly	(graphics.h)
floodfill	(graphics.h)
getarccoords	(graphics.h)
getaspectratio	(graphics.h)
getbkcolor	(graphics.h)
getcolor	(graphics.h)
getdefaulpalette	(graphics.h)
getdrivername	(graphics.h)
getfillpattern	(graphics.h)
getfillsettings	(graphics.h)
getgraphmode	(graphics.h)
getimage	(graphics.h)
getlinesettings	(graphics.h)
getmaxcolor	(graphics.h)
getmaxmode	(graphics.h)
getmaxx	(graphics.h)
getmaxy	(graphics.h)
getmodename	(graphics.h)
getmoderange	(graphics.h)
getpalette	(graphics.h)
getpalettesize	(graphics.h)
getpixel	(graphics.h)
gettextsettings	(graphics.h)
getviewsettings	(graphics.h)
getx	(graphics.h)
gety	(graphics.h)
graphefaults	(graphics.h)
grapherrormsg	(graphics.h)
graphresult	(graphics.h)
imagesize	(graphics.h)
initgraph	(graphics.h)
installuserdriver	(graphics.h)
installuserfont	(graphics.h)
line	(graphics.h)

linerel	(graphics.h)
lineto	(graphics.h)
moverel	(graphics.h)
moveto	(graphics.h)
outtext	(graphics.h)
outtextxy	(graphics.h)
pieslice	(graphics.h)
putimage	(graphics.h)
putpixel	(graphics.h)
rectangle	(graphics.h)
registerbgidriver	(graphics.h)
registerbgifont	(graphics.h)
restorecrtmode	(graphics.h)
sector	(graphics.h)
setactivepage	(graphics.h)
setallpalette	(graphics.h)
setaspectratio	(graphics.h)
setbkcolor	(graphics.h)
setcolor	(graphics.h)
setcursortype	(conio.h)
setfillpatttern	(graphics.h)
setfillstyle	(graphics.h)
setgraphbufsize	(graphics.h)
setgraphmode	(graphics.h)
setlinestyle	(graphics.h)
setpalette	(graphics.h)
setrgbpalette	(graphics.h)
settextjusttify	(graphics.h)
settextstyle	(graphics.h)
setusercharsize	(graphics.h)
setviewport	(graphics.h)
setvisualpage	(graphics.h)
setwritemode	(graphics.h)
textheight	(graphics.h)
textwidth	(graphics.h)
_graphfreemem	(graphics.h)
_graphgetmem	(graphics.h)

2.2.6 输入 / 输出例程

下面这些例程提供流级及 DOS 级输入 / 输出功能。

access	(io.h)
cgets	(conio.h)
chmod	(io.h)
chsize	(io.h)
clearerr	(stdio.h)
close	(io.h)
cprintf	(conio.h)
cputs	(conio.h)
creat	(io.h)
creatnew	(io.h)
creattemp	(io.h)
cscanf	(conio.h)
dup	(io.h)
dup2	(io.h)
eof	(io.h)
fclose	(stdio.h)
fcloseall	(stdio.h)
fdopen	(stdio.h)
feof	(stdio.h)
ferror	(stdio.h)
fflush	(stdio.h)
fgetc	(stdio.h)
fgetchar	(stdio.h)
fgetpos	(stdio.h)
fgets	(stdio.h)
filelength	(io.h)
fileno	(stdio.h)
flushall	(stdio.h)
fopen	(stdio.h)
fprintf	(stdio.h)
fputc	(stdio.h)
fputchar	(stdio.h)
fputs	(stdio.h)
fread	(stdio.h)
freopen	(stdio.h)
fscanf	(stdio.h)
fseek	(stdio.h)
fsetpos	(stdio.h)
fstat	(sys\stat.h)
ftell	(stdio.h)
fwrite	(stdio.h)
getc	(stdio.h)
getch	(conio.h)
getchar	(stdio.h)
getche	(conio.h)
getftime	(io.h)
getpass	(conio.h)
gets	(stdio.h)
getw	(stdio.h)
ioctl	(io.h)
isatty	(io.h)
kbhit	(conio.h)
lock	(io.h)
lseek	(io.h)
open	(io.h)
perror	(stdio.h)
printf	(stdio.h)
putc	(stdio.h)
putch	(conio.h)
putchar	(stdio.h)
puts	(stdio.h)
putw	(stdio.h)
read	(io.h)
remove	(stdio.h)
rename	(stdio.h)
rewind	(stdio.h)
scanf	(stdio.h)
setbuf	(stdio.h)
setcursortype	(conio.h)
setftime	(io.h)
setmode	(io.h)
setvbuf	(stdio.h)

sopen	(io.h)
sprintf	(stdio.h)
sscanf	(stdio.h)
stat	(sys\stat.h)
strerror	(stdio.h)
tell	(io.h)
tmpfile	(stdio.h)
tmpnam	(stdio.h)
ungetc	(stdio.h)
ungetch	(conio.h)
unlock	(io.h)
vfprintf	(stdio.h)
vfscanf	(stdio.h)
vprintf	(stdio.h)
vscanf	(stdio.h)
vsprintf	(stdio.h)
vsscanf	(io.h)
__chmod	(io.h)
__close	(io.h)
__creat	(io.h)
__open	(io.h)
__read	(io.h)
__strerror	(string.h, stdio.h)
__write	(io.h)

2.2.7 接口例程（DOS、8086、BIOS）

下面这些例程提供 DOS、BIOS 和与机器有关的功能。

absread	(dos.h)
abswrite	(dos.h)
bdos	(dos.h)
bdosptr	(dos.h)
bioscom	(bios.h)
biosdisk	(bios.h)
biosequip	(bios.h)
bioskey	(bios.h)
biosmemory	(bios.h)
biosprint	(bios.h)
biostime	(bios.h)
country	(dos.h)
ctrlbrk	(dos.h)
disable	(dos.h)
dosexterr	(dos.h)
enable	(dos.h)
FP__OFF	(dos.h)
FP__SEG	(dos.h)
freemem	(dos.h)
geninterrupt	(dos.h)
getcbrk	(dos.h)
getdfree	(dos.h)
getdta	(dos.h)
getfat	(dos.h)
getfatd	(dos.h)
getpsp	(dos.h)
getvect	(dos.h)
getverify	(dos.h)
harderr	(dos.h)
hardresume	(dos.h)
hardretn	(dos.h)
inport	(dos.h)
inportb	(dos.h)
int86	(dos.h)
int86x	(dos.h)
intdos	(dos.h)
intdosx	(dos.h)
intr	(dos.h)
keep	(dos.h)
MK__FP	(dos.h)
outport	(dos.h)
outportb	(dos.h)
parsfnm	(dos.h)
peek	(dos.h)
peekb	(dos.h)
poke	(dos.h)
pokeb	(dos.h)
randbrd	(dos.h)

randbwr	(dos.h)	setvect	(dos.h)
segread	(dos.h)	setverify	(dos.h)
setcbrk	(dos.h)	sleep	(dos.h)
setdta	(dos.h)	unlink	(dos.h)

2.2.8 操纵例程

下面这些例程处理字符串和内存块：拷贝、比较、移动和搜索。

memccpy	(mem.h, string.h)	strerror	(string.h)
memchr	(mem.h, string.h)	stricmp	(string.h)
memcmp	(mem.h, string.h)	strlen	(string.h)
memcpy	(mem.h, string.h)	strlwr	(string.h)
memicmp	(mem.h, string.h)	strncat	(string.h)
memmove	(mem.h, string.h)	strncmp	(string.h)
memset	(mem.h, string.h)	strncmpi	(string.h)
movedata	(mem.h, string.h)	strncpy	(string.h)
movemem	(mem.h, string.h)	strnicmp	(string.h)
setmem	(mem.h)	strnset	(string.h)
stpcpy	(string.h)	strpbrk	(string.h)
strcat	(string.h)	strrchr	(string.h)
strchr	(string.h)	strrev	(string.h)
strcmp	(string.h)	strset	(string.h)
strcmpi	(string.h)	strspn	(string.h)
strcoll	(string.h)	strstr	(string.h)
strcpy	(string.h)	strtok	(string.h)
strcspn	(string.h)	strupr	(string.h)
strdup	(string.h)	strxfrm	(string.h)

2.2.9 数学例程

下面这些例程执行数学计算和转换。

abs	(complex.h,stdlib.h)	cabs	(math.h)
acos	(complex.h,math.h)	ceil	(math.h)
arg	(complex.h)	complex	(complex.h)
asin	(complex.h,math.h)	conj	(conplex.h)
atan	(complex.h,math.h)	cos	(complex.h,math.h)
atan2	(complex.h,math.h)	cosh	(complex.h,math.h)
atof	(stdlib.h,math.h)	div	(math.h)
atoi	(stdlib.h)	ecvt	(stdlib.h)
atol	(stdlib.h)	exp	(math.h)
bcd	(bcd.h)	fabs	(math.h)