

· 节省脑部存储空间 ·

脑动力

C语言函数 速查效率手册

韩先锋◎编著



11小时多媒体语音视频讲解

全面、实用、查询方便的C语言必备工具
343个函数、343个示例、40个技巧



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

脑动力

C语言函数

速查效率手册

韩先锋◎编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

前 言

在计算机发展的历史上，没有哪一种程序设计语言像 C 语言这样应用如此广泛。C 语言是一种高级程序设计语言，是 20 世纪 70 年代由 Dennis Ritchie 和 Brian Kernighan 在贝尔实验室开发的。虽然 C 语言最初是作为一种系统语言设计的，但后来的实践证明 C 语言功能强大，也十分灵活，可以用于各种应用程序，如商业软件、工程项目等。C 语言在个人计算机编程领域非常流行，因为 C 语言规模较小——比其他语言需求的内存少。

本书的特点

与其他大部分语言相比，C 语言是一种非常优秀的程序设计入门语言，而且其简洁易学。为了便于程序员编程，提高编程的质量及提高程序的运行效率，在编程的时候应该尽可能多地使用库函数。例如为了显示一段文字，我们在 C 语言中也找不到显示语句，只能使用库函数 `printf`。如果要直接使用 C 语言的语句计算 `sin` 或 `cos`，会造成编写程序困难，但是函数库提供了 `sin` 和 `cos` 函数，所以编程就方便多了。本书将以初学者的视角介绍基本库函数。

本书的主要特点如下：

- 函数全面，内容翔实。C 语言有着众多的函数库，函数数量众多，笔者收集了尽可能多的函数进行详细的讲解，

这些函数覆盖了常见的各个领域，基本能满足读者的各项需求。

- 分类清晰，易于查询。本书的目录将函数按功能分类，每一节都有清晰的功能介绍，便于读者查询。
- 理论讲解与实例分析相结合。理论讲解通学内容全面但不够生动，实例分析通学能使用户印象深刻，但全面性尚有不足。本书将两者有机地结合起来，实例分析与理论讲解并重，使函数更易掌握。
- 注重基础，讲解细致。书中对每个函数都进行了详细的解释，包括函数原型、函数功能、参数说明、示例详解等，让读者能够迅速地掌握每个函数的使用方法。
- 实例讲解。本书对于每个函数都配有一个简单明了的例子，这些例子直观地展示了函数的使用方法，同时每个例子下面都配有运行结果及程序的详细解释，让读者可以参考示例进行实践。本书的所有例子、源代码都附在随书光盘中，方便读者使用。

本书的主要内容

第 1 章的内容主要是 C 语言基础知识介绍。首先介绍了 C 语言的发展史，随后从 C 语言的两个主要的开发环境 Turbo C 2.0 和 Microsoft Visual C++ 6.0 简单介绍如何进行代码的编写及简单的调试。

第 2 章介绍分类函数。ASCII 字符可以分为英文字母、数字、控制字符、空白字符、大小写字母及标点符号，分类是指对字符进行属性判定，判断字符属于哪个范畴，这些属性的判定在程序中非常常见，在通信协议的字符处理部分更是必不可少。

第 3 章介绍字符串操作函数。本章的字符串操作包括字符串的复制、连接、检索、比较、转换、置位等各种操作。

第 4、5 章是输入/输出函数。输入/输出是指程序与环境或者用户之间进行的数据或者信息交换，程序必须要有输入/输出，不然实际作用就不会太大，同样用户也需要通过输入提供程序必要的初始化数据，控制程序按照期望的方式运行，程序通过输出对外部环境产生某种影响作为程序运行的结果。

第 6 章介绍数学函数。本章将介绍数学函数的各种常见函数，比如三角函数、求绝对值、幂函数、指数函数、多项式、随机函数等。

第 7 章介绍时间函数。主要是处理日期和时间的函数，包括时间格式的转换及计算机时间的读取和设置。

第 8 章介绍转换函数。转换函数是指将不同类型的数据进行格式转换，以满足实际使用需要。C 语言函数库提供了一些基本的转换函数，包括数字转字符串、字符串转数字，同时也包括大小写转换和编码转换。这些都是在实际编程过程中经常用到的函数。

第 9、10 章介绍图形函数。计算机图形是由像素表示的，一个像素通常被视为图像的最小完整采样。本章介绍的图形函数有点线型函数、填充函数、字体函数及其他图形函数集合。

第 11 章是诊断函数。诊断函数是 C 语言函数库中提供的用于判断程序语句中某些条件是否与预计的结果一样，或者根据已有错误情况提供的一些错误信息供使用者更好地改正错误。C 语言中诊断函数不多，但却非常实用，一名优秀的程序员会习惯性地使用诊断函数规整自己的程序行为。

第 12 章介绍接口函数。这部分内容读者可能会比较陌生，本章主要讲解 dos.h 头文件中的函数，内容包括操作系统 BIOS、DOS 等底层介质，对磁盘的读写也是很重要的部分，包括磁盘扇区信息、文件分配表信息等。做过嵌入式开发的读者还会发现其中还有关于中断的各种操作函数。

第 13 章介绍目录函数。目录函数主要包括一些文件、目录相关的操作。本章包括的目录函数就是文件名操作、目录相关操作及创建和删除等。

第 14 章介绍进程函数。进程是一个具有一定独立功能的程序，也是关于某个数据集合的一次运行活动。它是操作系统动态执行的基本单元，本章将介绍进程的创建、中止等操作函数。

第 15 章介绍内存函数。内存是计算机中所有程序运行的地方，它还是与 CPU 进行沟通的桥梁，因此内存的操作是计算机中很重要的部分。本章分两部分介绍空间地址和内存数据。

本书以介绍 C 语言函数为主，是一本函数查询的参考书。本书详细地介绍了常用的 C 语言函数，重点介绍每个函数的参数意义、返回值的意义及函数使用时的注意事项。本书包含大量实例，详细讲解每个函数的使用方法，每个函数均给出了对应的应用实例，方便读者理解函数并进行模拟实践。

适合阅读本书的读者

- C 语言初学者。没有基础，希望学习程序设计的新手。
- 大中专院校的学生。学生尚处于学习阶段，常常需要查询 C 语言函数的用法。
- 研发人员。具备一定的编程理论知识，但是缺乏实践操作的初级程序人员或者从其他语言转向学习 C 语言程序设计的中级编程人员。本书能够帮助研发人员及时找到常用函数的正确使用方法。

编 者

目 录

第1章 C语言基础	1
1.1 C语言概述	1
1.2 C语言开发环境 Turbo C 2.0 的基础知识	4
1.2.1 File (文件) 菜单	5
1.2.2 Edit (编辑) 菜单	6
1.2.3 Run (运行) 菜单	7
1.2.4 Compile (编译) 菜单	8
1.2.5 Project (项目) 菜单	9
1.2.6 Options (选项) 菜单	10
1.2.7 Debug (调试) 菜单	11
1.2.8 Break/watch (断点及监视表达式) 菜单	11
1.3 使用 Turbo C 运行程序	12
1.4 使用 Microsoft Visual C++ 6.0 运行程序	13
第2章 分类函数	16
2.1 字母数字	16
2.1.1 判断字符是否为英文字母 isalpha()	16
2.1.2 判断字符是否为数字 isdigit()	17
2.1.3 判断字符是否为英文字母或数字 isalnum()	18
2.1.4 判断字符是否为小写字母 islower()	19
2.1.5 判断字符是否为大写字母 isupper()	20
2.2 字符	21
2.2.1 判断字符是否为 ASCII 码 isascii()	21
2.2.2 判断字符是否为控制字符 isctrl()	22
2.2.3 判断字符是否为可打印字符 (不含空格) isgraph()	23

2.2.4	判断字符是否为可打印字符（含空格） isprint()	24
2.2.5	判断字符是否为标点符号 ispunct()	25
2.2.6	判断字符是否为空白字符 isspace()	26
2.2.7	判断字符是否为十六进制字符 isxdigit()	27
第 3 章	字符串操作函数	29
3.1	复制	29
3.1.1	复制字符串到指定的数组中 strcpy()	29
3.1.2	复制字符串到指定的数组中 strncpy()	30
3.1.3	复制字符串到最近申请的单元 strdup()	31
3.1.4	复制前 N 个字符 strncpy()	32
3.2	连接	33
3.2.1	把一个字符串添加到另一个字符串的 结尾处 strcat()	33
3.2.2	连接指定个数的字符到另一个字符串的 结尾处 strncat()	34
3.3	检索	35
3.3.1	检索某字符首次出现的位置 strchr()	35
3.3.2	查找字符串中最后一次出现某字符的位置 strrchr()	36
3.3.3	在一个字符串中搜索另一个字符串中的 字符 strcspn()	37
3.3.4	求两个字符串中首个相同字符的位置 strpbrk()	37
3.3.5	求字符串中第一个不在指定字符串中出现 字符的下标 strspn()	38
3.3.6	求子字符串首次出现的位置 strstr()	39
3.4	比较	40
3.4.1	比较字符串（区分大小写）strcmp()	40

3.4.2	比较字符串 (不分大小写) <code>strcmp()</code>	41
3.4.3	比较字符串的前几位字符 (区分大小写) <code>strncmp()</code>	42
3.4.4	比较字符串的前几位字符 (不分大小写) <code>strnicmp()</code>	43
3.4.5	求字符串长度 <code>strlen()</code>	44
3.5	转换	45
3.5.1	将字符串转换为小写 <code>strlwr()</code>	45
3.5.2	将字符串转换为大写 <code>strupr()</code>	46
3.5.3	反转字符串 <code>strrev()</code>	46
3.5.4	分解字符串 <code>strtok()</code>	47
3.6	置位	48
3.6.1	将字符串前 <i>n</i> 个字符置位 <code>strnset()</code>	48
3.6.2	将字符串设置为指定的字符 <code>strset()</code>	49
第 4 章	输入/输出函数 (一)	50
4.1	键盘控制台	50
4.1.1	当前键盘输入 <code>kbhit()</code>	50
4.1.2	从控制台 (键盘) 读取一个字符 <code>getch()</code>	51
4.1.3	从控制台 (键盘) 读取一个字符, 并显示在 屏幕上 <code>getchar()</code>	52
4.1.4	从控制台 (键盘) 读取一个字符, 并显示在 屏幕上 <code>getche()</code>	53
4.1.5	从流中读取一个字符 <code>fgetchar()</code>	54
4.1.6	向控制台写一个字符 <code>putch()</code>	55
4.1.7	输出一个字符到控制台 <code>putchar()</code>	55
4.1.8	把字符退回到键盘缓冲区 <code>ungetch()</code>	56
4.1.9	把一个字符退回输入流 <code>ungetc()</code>	57
4.1.10	从控制台 (键盘) 读取一个字符串 <code>cgets()</code>	58
4.1.11	格式化输入 <code>scanf()</code>	59

4.1.12	从 stdin 中执行格式化输入 vscanf()	60
4.1.13	格式化输入 cscanf()	61
4.1.14	从字符串中读取指定格式的数据 sscanf()	62
4.1.15	从 stdin 中执行格式化输入 vsscanf()	62
4.1.16	将一个字符串放入输出流中 puts()	63
4.1.17	格式化输出 printf()	64
4.1.18	格式化输出 vprintf()	65
4.1.19	格式化输出至屏幕 cprintf()	66
4.1.20	将格式化的数据写入字符串 sprintf()	67
4.1.21	将格式化的数据放入字符串中 vsprintf()	68
4.2	设备	69
4.2.1	控制 I/O 设备 ioctl	69
4.2.2	设置某一信号的相应动作 signal()	71
4.2.3	向正在执行的程序发送一个信号 raise()	72
第 5 章	输入/输出函数 (二)	74
5.1	文件	74
5.1.1	修改文件名 rename()	74
5.1.2	打开文件 open()	75
5.1.3	新建文件 creat()	77
5.1.4	创建一个新文件 creatnew()	77
5.1.5	从文件中读 read()	78
5.1.6	往已经打开的文件中写数据 write()	80
5.1.7	复制文件句柄 dup()	81
5.1.8	复制文件句柄 dup2()	82
5.1.9	检测文件结束 eof()	84
5.1.10	移动文件读/写指针 lseek()	85
5.1.11	获取文件指针的当前位置 tell()	86
5.1.12	获取设备类型 isatty()	87
5.1.13	设置文件共享锁 lock()	88

5.1.14	解除文件共享锁 unlock()	89
5.1.15	关闭文件 close()	90
5.1.16	打开一个文件 fopen()	91
5.1.17	从流中读取字符 getc()	93
5.1.18	往指定流中输入一个字符 putc()	94
5.1.19	从流中读取整数 getw()	95
5.1.20	向流中写整数 putw()	97
5.1.21	把一个字符退回到输入流中 ungetc()	98
5.1.22	送一个字符到流中 fputc()	99
5.1.23	从流中读取一个字符 fgetc()	100
5.1.24	从流中读取一行或者指定个数的字符 fgets()	101
5.1.25	送一个字符串到流中 fputs()	102
5.1.26	从一个流中读取数据 fread()	103
5.1.27	向流中写入一个数据块 fwrite()	105
5.1.28	从一个流中执行格式化输入 fscanff()	106
5.1.29	将格式化数据送到流中 fprintff()	107
5.1.30	从流中执行格式化输入 vfscanff()	108
5.1.31	往流中格式化输出 vfprintff()	109
5.1.32	移动文件的读/写指针到指定的位置 fseek()	111
5.1.33	获取文件读/写指针的当前位置 ftell()	112
5.1.34	将文件的读/写指针重新指向开头 rewind()	113
5.1.35	检查流上的文件结束标识 feof()	114
5.1.36	检测文件流是否出错 ferror()	116
5.1.37	复位流错误标识 clearerr()	117
5.1.38	把缓冲区与流相连 setbuf()	118
5.1.39	为流指定特殊的缓冲区 setvbuf()	119
5.1.40	关闭文件流 fclose()	120
5.1.41	关闭所有打开的流 fcloseall()	121
5.1.42	清空文件缓冲区 fflush()	122

5.1.43	往已经打开的文件中写数据 write().....	122
5.2	文件属性	124
5.2.1	设置文件属性 chmod().....	124
5.2.2	获取文件属性 access().....	125
5.2.3	获取文件的日期与时间 getftime().....	126
5.2.4	设置文件的日期与时间 setftime()	127
5.2.5	获取文件长度 filelength().....	128
5.2.6	设置打开文件方式 setmode().....	129
第 6 章	数学函数.....	131
6.1	绝对值	131
6.1.1	整数的绝对值 abs()	131
6.1.2	求双精度数据的绝对值 fabs()	132
6.1.3	求长整型数的绝对值 labs()	133
6.2	三角函数	134
6.2.1	正弦函数 sin()	134
6.2.2	反正弦函数 asin()	135
6.2.3	余弦函数 cos()	136
6.2.4	反余弦函数 acos().....	136
6.2.5	正切函数 tan().....	137
6.2.6	反正切函数 atan()	138
6.2.7	反正切函数 atan2()	138
6.2.8	求直角三角形的斜边长 hypot ().....	139
6.3	其他	140
6.3.1	计算多项式 ploy().....	140
6.3.2	将双精度数分解为整数和小数部分 modf().....	141
6.3.3	求余数 fmod()	142
6.3.4	把浮点数分解为尾数和指数 frexp().....	142
6.3.5	指数函数运算 exp()	143
6.3.6	计算 value 乘以 2 的 exp 次幂 ldexp().....	144

6.3.7	计算以 e 为底的对数值 <code>log()</code>	145
6.3.8	计算以 10 为底的对数值 <code>log10()</code>	145
6.3.9	计算 x 的 y 次幂 <code>pow()</code>	146
6.3.10	计算 10 的 x 次幂 <code>pow10()</code>	147
6.3.11	求一个非负数的平方根 <code>sqrt()</code>	147
6.3.12	求双曲余弦值 <code>cosh()</code>	148
6.3.13	求双曲正弦值 <code>sinh()</code>	149
6.3.14	求双曲正切值 <code>tanh()</code>	149
6.3.15	求大于或者等于指定数的最小整数 <code>ceil()</code>	150
6.3.16	求小于或者等于指定数的最大整数 <code>floor()</code>	151
6.3.17	随机数发生器的初始化函数 <code>srand()</code>	151
6.3.18	伪随机数发生器 <code>rand()</code>	152
第 7 章	时间函数	154
7.1	转换时间格式	154
7.1.1	把日期和时间转换为字符串 <code>ctime()</code>	154
7.1.2	把时间转换为 ASCII 码 <code>asctime()</code>	155
7.1.3	把时间转换为格林尼治时间 <code>gmtime()</code>	156
7.1.4	把时间转换为当地标准时间 <code>localtime()</code>	157
7.1.5	UNIX 时间兼容函数 <code>tzset()</code>	158
7.1.6	把时间转换为 UNIX 格式 <code>dostounix()</code>	158
7.1.7	把时间转换为 DOS 格式 <code>unixtodos()</code>	159
7.2	读取和设置计算机时间	160
7.2.1	获取计算机日期 <code>getdate()</code>	161
7.2.2	设置计算机日期 <code>setdate()</code>	161
7.2.3	获取计算机时间 <code>gettime()</code>	162
7.2.4	设置计算机时间 <code>settime()</code>	163
7.2.5	获取当前系统时间 <code>time()</code>	164
7.2.6	设置当前系统时间 <code>stime()</code>	164
7.2.7	计算时间差 <code>difftime()</code>	165

第 8 章 转换函数	167
8.1 数字转字符串	167
8.1.1 将双精度数转换为字符串 <code>ecvt()</code>	167
8.1.2 将浮点数转换为字符串 <code>fcvt()</code>	168
8.1.3 将浮点数转换为字符串 <code>gcvt()</code>	169
8.1.4 将无符号长整型数据转换为任意进制的字符串 <code>ultoa()</code>	170
8.1.5 将长整型数据转换为字符串 <code>ltoa()</code>	171
8.1.6 将整数转换为字符串 <code>itoa()</code>	172
8.2 字符串转数字	173
8.2.1 将字符串转为浮点数 <code>atof()</code>	173
8.2.2 将字符串转换为整数 <code>atoi()</code>	173
8.2.3 将字符串转换为长整型数据 <code>atol()</code>	174
8.2.4 将字符串转换为浮点数 <code>strtod()</code>	175
8.2.5 将字符串转换成长整型数据 <code>strtol()</code>	176
8.3 其他转换函数	177
8.3.1 将字符转换为小写字母 <code>tolower()</code>	178
8.3.2 将字符转换成大写字母 <code>toupper()</code>	178
8.3.3 求字符对应的 ASCII 码 <code>toascii()</code>	179
第 9 章 图形函数（一）	181
9.1 点线型函数	181
9.1.1 获取指定位置像素的颜色 <code>getpixel()</code>	182
9.1.2 在指定位置绘制像素 <code>putpixel()</code>	184
9.1.3 在指定的两点之间画一直线 <code>line()</code>	185
9.1.4 从当前点到有一定相对距离的点画直线 <code>linereel()</code>	187
9.1.5 设置当前线的宽度和类型 <code>setlinestyle()</code>	189
9.1.6 获取可以传递给函数 <code>setcolor</code> 的最大颜色值 <code>getmaxcolor()</code>	191

9.1.7	设置当前画线颜色 setcolor()	193
9.1.8	获取当前画线颜色 getcolor()	195
9.1.9	获取当前线型信息 getlinesettings()	196
9.1.10	绘制矩形 rectangle()	198
9.1.11	绘制多边形 drawpoly()	199
9.1.12	绘制多边形并填充 fillpoly()	201
9.1.13	获取最后一次调用 arc 的坐标 getarcoords()	203
9.1.14	绘制圆弧 arc()	205
9.1.15	绘制圆 circle()	206
9.1.16	绘制扇形并填充 pieslice()	208
9.1.17	绘制椭圆 ellips()	209
9.1.18	绘制椭圆并填充 fillellipse()	211
9.2	填充函数	212
9.2.1	获取当前的填充模式和填充颜色 getfillsettings()	213
9.2.2	将用户定义的填充模式复制到内存中 getfillpattern()	215
9.2.3	选择用户定义的填充模式 setfillpattern()	217
9.2.4	设置填充模式和颜色 setfillstyle()	218
9.2.5	获取当前图形模式的纵横比 getaspectratio()	220
9.2.6	设置图形模式的纵横比 setaspectratio()	222
9.2.7	获取当前背景颜色 getbkcolor()	224
9.2.8	设置背景颜色 setbkcolor()	226
9.2.9	获取调色板信息 getpalette()	228
9.2.10	设置调色板的颜色 setpalette()	230
9.2.11	设置调色板所有颜色值 setallpalette()	232
9.2.12	填充一个有界区域 floodfill()	234

第 10 章 图形函数（二）	237
10.1 字体函数	237
10.1.1 设置图形输出系统的文本类型 settextstyle().....	237
10.1.2 获取当前图形文本字体的信息 gettextsettings().....	239
10.1.3 在当前点输出字符串 outtext().....	241
10.1.4 在指定位置显示一个字符串 outtextxy().....	243
10.1.5 设置图形函数的文本对齐方式 settextjustify().....	244
10.1.6 设置矢量字体的宽度和高度 setusercharsize().....	246
10.1.7 获取以像素为单位的字符串高度 textheight().....	247
10.1.8 获取以像素为单位的字符串宽度 textwidth().....	249
10.1.9 选择低亮度字符 lowvideo().....	251
10.1.10 选择高亮度字符 highvideo().....	251
10.2 其他函数大集合	252
10.2.1 初始化图形系统 initgragh()	252
10.2.2 通过检查硬件确定图形驱动程序和 模式 detectgraph()	254
10.2.3 获取当前视窗信息 getviewsettings().....	255
10.2.4 获取当前图形位置的 X 坐标 getx()	257
10.2.5 获取当前图形位置的 Y 坐标 gety()	258
10.2.6 设置当前视窗 setviewport()	260
10.2.7 清空视窗 clearviewport().....	262
10.2.8 清除图形屏幕 cleardevice().....	263
10.2.9 获取屏幕的最大 X 坐标 getmaxx()	265
10.2.10 获取屏幕的最大 Y 坐标 getmaxy()	266

10.2.11	移动当前点到指定位置 <code>moveto()</code>	268
10.2.12	移动当前点到一相对距离 <code>moverel()</code>	269
10.2.13	绘制二维条形图 <code>bar()</code>	270
10.2.14	绘制三维条形图 <code>bar3d()</code>	272
10.2.15	关闭图形系统 <code>closegraph()</code>	273
10.2.16	设置内部图形缓冲区的大小 <code>setgraphbufsize()</code>	275
10.2.17	获取当前图形模式 <code>getgraphmode()</code>	276
10.2.18	获取图形模式名 <code>getmodename()</code>	278
10.2.19	设置当前图形模式 <code>setgraphmode()</code>	279
10.2.20	保存指定区域的图像到内存 <code>getimage()</code>	281
10.2.21	将内存中的图像数据显示到屏幕上 <code>putimage()</code>	284
10.2.22	获取给定图形驱动程序的模式范围 <code>getmoderange()</code>	287
10.2.23	把当前的图形系统设置都复位为默认值 <code>graphdefaults()</code>	288
10.2.24	将屏幕模式恢复为文本模式 <code>restorecrtmode()</code>	290
10.2.25	选择图形输出活动页 <code>setactivepage()</code>	292
10.2.26	设置可见图形活动页 <code>setvisualpage()</code>	294
10.2.27	获取最后一次错误的图形操作的错误代码 <code>graphresult()</code>	296
10.2.28	获取一个错误信息 <code>grapherrormsg()</code>	297
10.2.29	用户可修改的图形存储区分配函数 <code>_graphgetmem()</code>	298
10.2.30	用户可修改的图形存储区释放函数 <code>_graphfreemem()</code>	300