

◆-----实用掌中宝-----◆

涵盖352个C语言常用函数，提供了352个典型示例

C语言

常用函数速查手册

陈超 等编著

- ◎ 涵盖了C语言常用的21个函数库的最常用函数
- ◎ 从函数的功能、参数和返回值三方面讲解函数的使用
- ◎ 所有函数都给出了针对性的示例，加深读者的理解
- ◎ 所有的示例代码都给出了详细的注释和完整的讲解



化学工业出版社

◆-----实用掌中宝-----◆

C语言

常用函数速查手册



化学工业出版社

· 北京 ·

本书全面、系统地讲解了 C 语言相关的 21 个函数库,所涉及的函数多达 352 个。为了方便读者学习,每一个函数都依次对其作用、语法形式、参数、返回值进行了讲解。同时,每个函数都配有专门的例子,供读者参考学习。最后给出了本书所涉及 C 语言函数的索引,便于读者检索。

为了方便读者查找,所有函数都按照所在库进行分章讲解。这样既方便读者系统学习,也方便同类函数的对比和查找。本书所涉及的函数全面,适合所有想学习 C 语言的开发人员、爱好者和大中专院校学生使用。对于经常采用 C 语言进行开发的开发人员,更是一本不可多得的案头必备工具参考书。

图书在版编目 (CIP) 数据

C 语言常用函数速查手册 / 陈超等编著. —北京:
化学工业出版社, 2010.6

(实用掌中宝)

ISBN 978-7-122-08136-0

I. C… II. 陈… III. C 语言-程序设计-技术
手册 IV. TP312-62

中国版本图书馆 CIP 数据核字 (2010) 第 056114 号

责任编辑: 陈 静

装帧设计: 蓝色印象

责任校对: 宋 玮

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 北京市彩桥印刷有限责任公司

880mm×1230mm 1/32 印张 11¹/₂ 字数 325 千字

2010 年 6 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定价: 28.00 元

版权所有 违者必究

前 言

C 语言作为一门基础语言,深受广大编程爱好者的喜爱。由于其语法简单,功能强大,很多高校都把 C 语言作为必修课程,而初学者往往选择它作为编程的入门语言。同时,各类计算机考试的考察编程语言也为 C 语言,如全国计算机等级考试、程序员考试等。

C 语言语法简单,通过各类库函数,可以实现所有功能。由于 C 语言诞生较早,现在很少有厂商提供完善的 IDE 支持,造成学习 C 语言的障碍,尤其是了解 C 语言函数。编者在日常开发中,对此深有感触。为了方便大家学习,编者花费了半年时间整理 C 语言各类函数,完成此书。

一、本书特色

1. 精选 21 个函数库, 352 个函数

虽然 C 语言语法简单,但包含众多的函数库和函数。其数量众多,但是常用的比较有限。笔者根据多年 C 语言使用经验,精选了 21 个函数库,352 个函数进行讲解。这些函数覆盖 C 语言各个常见领域,基本满足读者的各项需要。

2. 讲解细致, 重点明确

为了方便读者学习,本书对每个函数都进行详细讲解,依次对函数的作用、语法、参数、返回值进行仔细描述,让读者可以快速掌握每个函数的使用。同时,书上出现的代码进行相应的压缩,避免过多的代码对读者的理解和使用造成影响。

3. 示例讲解, 便于实践

为了方便读者更深入理解,本书每个函数都配以一个示例,这些示例具有很强的代表性。读者通过这些示例,不仅可以更快了解函数的使用,还可以参照示例进行动手实践。

4. 讲解由浅入深，循序渐进，适合各个层次的读者阅读

本书从 C 语言最基本的函数开始讲解，逐步深入到 C 语言的其他函数，内容从易到难，讲解由浅入深，循序渐进，适合各个层次的读者阅读，并均有所获。

二、本书内容

第 1 章介绍了 C 语言的标准控制台输入输出函数库。通过对本章的学习，读者可以学习和掌握各种与输入输出操作相关函数的结构与使用。

第 2 章介绍了 C 语言的数学函数库。通过对本章的学习，读者可以掌握 C 语言中关于数学函数的语法与使用。

第 3 章介绍了 C 语言的字符函数库。通过对本章的学习，读者可以掌握 C 语言中关于字符函数的相关操作与使用。

第 4 章介绍了 C 语言的字符串函数库。通过对本章的学习，读者可以掌握 C 语言中关于字符串函数的操作与使用。

第 5 章介绍了 C 语言的标准库函数库。通过对本章的学习，读者可以掌握 C 语言中关于标准库函数的语法结构与使用。

第 6 章介绍了 C 语言的图形处理函数库。通过对本章的学习，读者可以掌握 C 语言中关于图形处理函数的语法结构与使用。

第 7 章介绍了 C 语言的动态内存管理函数库。通过对本章的学习，读者可以掌握 C 语言中关于动态内存管理函数的结构与使用。

第 8 章介绍了 C 语言的目录操作函数库。通过对本章的学习，读者可以掌握 C 语言中关于目录操作函数的结构与使用。

第 9 章介绍了 C 语言的系统接口函数库。通过对本章的学习，读者可以掌握 C 语言中关于系统接口函数的使用。

第 10 章介绍了 C 语言的输入输出函数库。通过对本章的学习，读者可以掌握 C 语言中关于输入输出函数的相关操作。

第 11 章介绍了 C 语言的浮点数据处理库。通过对本章的学习，读者可以掌握 C 语言中关于浮点数函数的用法。

第 12 章介绍了 C 语言的控制台输入输出函数库。通过对本章的学习，读者可以掌握 C 语言中关于控制台输入输出函数的操作与使用。

第 13 章介绍了 C 语言的 DEBUG 相关函数库。通过对本章的学习,读者可以掌握 C 语言中关于 DEBUG 函数的结构与使用。

第 14 章介绍了 C 语言的 BIOS 相关函数库。通过对本章的学习,读者可以掌握 C 语言中关于 BIOS 函数的结构与使用。

第 15 章介绍了 C 语言的内存相关函数库。通过对本章的学习,读者可以掌握 C 语言中关于内存相关函数的使用。

第 16 章介绍了 C 语言的进程管理函数库。通过对本章的学习,读者可以掌握 C 语言中关于进程管理相关函数的语法结构与使用。

第 17 章介绍了 C 语言的函数跳转函数库。通过对本章的学习,读者可以掌握 C 语言中关于函数跳转的使用。

第 18 章介绍了 C 语言的信号定义函数库。通过对本章的学习,读者可以掌握 C 语言中关于信号定义函数的结构与使用。

第 19 章介绍了 C 语言的函数参数处理函数库。通过对本章的学习,读者可以掌握 C 语言中关于函数参数处理函数的结构与使用。

第 20 章介绍了 C 语言的时间函数库。通过对本章的学习,读者可以掌握 C 语言中关于时间函数的应用。

第 21 章介绍了 C 语言的标准工具库函数库。通过对本章的学习,读者可以掌握 C 语言中关于标准工具库函数的使用。

索引:按英文字母顺序列出本书所涉及函数的索引,便于读者检索。

三、本书读者

- ☐ C 语言初学者。
- ☐ C 语言专业开发人员。
- ☐ 大中专院校的学生。
- ☐ 社会培训班学员。
- ☐ C 语言相关程序员。

四、本书编者

本书主要由陈超编写,其他参与编写和资料整理的人员有刘成、马臣云、潘娜、阮履学、陶则熙、王大强、王磊、徐琦、许少峰、颜

编者

2010年3月

目 录

第 1 章 输入输出函数库: <code>stdio.h</code>	1
1.1 从流中取字符函数: <code>getc()</code>	1
1.2 从 <code>stdin</code> 流中读字符函数: <code>getchar()</code>	2
1.3 从控制台取字符 (带回显) 函数: <code>getche()</code>	2
1.4 从流中取整数函数: <code>getw()</code>	3
1.5 输出一个字符到指定流中函数: <code>putc()</code>	4
1.6 在 <code>stdout</code> 上输出字符函数: <code>putchar()</code>	5
1.7 将字符串送到流中函数: <code>puts()</code>	6
1.8 从流中取字符函数: <code>gets()</code>	7
1.9 将字符或字送到流中函数: <code>putw()</code>	7
1.10 重命名文件函数: <code>rename()</code>	9
1.11 删除文件函数: <code>remove()</code>	9
1.12 将文件指针重新指向流的开头函数: <code>rewind()</code>	10
1.13 系统错误信息: <code>perror()</code>	11
1.14 把缓冲区与流相联函数: <code>setbuf()</code>	12
1.15 把缓冲区与流相关函数: <code>setvbuf()</code>	13
1.16 将格式化输出到字符串中函数: <code>sprintf()</code>	13
1.17 执行字符串中的格式化输入函数: <code>sscanf()</code>	14
1.18 以二进制方式打开暂存文件函数: <code>tmpfile()</code>	16
1.19 创建一个唯一的文件名函数: <code>tmpnam()</code>	17
1.20 把字符退回到输入流函数: <code>ungetc()</code>	17
1.21 把字符退回到键盘缓冲区函数: <code>ungetch()</code>	18
1.22 取得当前文件的句柄函数: <code>fgetpos()</code>	19
1.23 定位流上的文件指针函数: <code>fsetpos()</code>	20
1.24 打开流函数: <code>fopen()</code>	21
1.25 关闭流函数: <code>fclose()</code>	22
1.26 清除流函数: <code>fflush()</code>	22
1.27 检测流上的错误函数: <code>ferror()</code>	23

1.28	替换流函数: <code>freopen()</code>	24
1.29	复位错误标志函数: <code>clearerr()</code>	25
1.30	从流中读取字符函数: <code>fgetc()</code>	26
1.31	从流中读取字符函数: <code>fgetchar()</code>	27
1.32	输出字符到标准输出流 (<code>stdout</code>) 函数: <code>fputchar()</code>	27
1.33	从流中读取字符函数: <code>fgets()</code>	28
1.34	检测流上的文件结束符函数: <code>feof()</code>	29
1.35	送字符串到流中函数: <code>fputs()</code>	30
1.36	从流中读数据函数: <code>fread()</code>	30
1.37	写内容到流函数: <code>fwrite()</code>	31
1.38	将格式化内容输出到流函数: <code>fprintf()</code>	32
1.39	从流中执行格式化输入内容函数: <code>fscanf()</code>	33
1.40	格式化输入函数: <code>scanf()</code>	34
1.41	格式化输出函数: <code>printf()</code>	36
1.42	重定位流上的文件指针函数: <code>fseek()</code>	37
1.43	返回当前文件指针函数: <code>ftell()</code>	38
第 2 章	数学函数库: <code>math.h</code>	39
2.1	浮点数绝对值函数: <code>fabs()</code>	39
2.2	整数绝对值函数: <code>abs()</code>	40
2.3	反余弦值函数: <code>acos()</code>	40
2.4	反正弦值函数: <code>asin()</code>	41
2.5	反正切函数: <code>atan()</code>	42
2.6	Y/X反正切函数: <code>atan2()</code>	42
2.7	不小于某数的最小整数函数: <code>ceil()</code>	43
2.8	余弦值函数: <code>cos()</code>	44
2.9	双曲余弦值函数: <code>cosh()</code>	44
2.10	e的次幂函数: <code>exp()</code>	45
2.11	不大于某数的最大整数函数: <code>floor()</code>	46
2.12	余数函数: <code>fmod()</code>	47
2.13	计算浮点数尾数和指数函数: <code>frexp()</code>	47
2.14	计算幂函数: <code>ldexp()</code>	48
2.15	取浮点数小数部分函数: <code>modf()</code>	49
2.16	计算直角三角形斜边长度函数: <code>hypot()</code>	50

2.17	自然对数函数: <code>log()</code>	51
2.18	对数函数: <code>log10()</code>	51
2.19	计算x的y次幂函数: <code>pow()</code>	52
2.20	计算 10 的x次幂函数: <code>pow10()</code>	53
2.21	正弦值函数: <code>sin()</code>	53
2.22	双曲正弦值函数: <code>sinh()</code>	54
2.23	平方根函数: <code>sqrt()</code>	55
2.24	正切值函数: <code>tan()</code>	55
2.25	双曲正切值函数: <code>tanh()</code>	56
第 3 章	字符函数库: <code>ctype.h</code>	57
3.1	判断字符是否为控制字符函数: <code>isctrl()</code>	57
3.2	判断字符是否为字母或数字函数: <code>isalnum()</code>	58
3.3	判断字符是否为英文字母函数: <code>isalpha()</code>	59
3.4	判断字符是否为ASCII码函数: <code>isascii()</code>	60
3.5	判断字符是否为TAB或空格函数: <code>isblank()</code>	61
3.6	判断字符是否为除空格外的可打印字符函数: <code>isgraph()</code>	61
3.7	判断字符是否为小写英文字母函数: <code>islower()</code>	62
3.8	判断字符是否为可打印字符(含空格)函数: <code>isprint()</code>	63
3.9	判断字符是否为标点符号函数: <code>ispunct()</code>	64
3.10	判断字符是否为空白符函数: <code>isspace()</code>	65
3.11	判断字符是否为大写英文字母函数: <code>isupper()</code>	66
3.12	判断字符是否为十六进制数字函数: <code>isxdigit()</code>	67
3.13	将字符转换为ASCII码函数: <code>toascii()</code>	68
3.14	将字符转换为小写英文字母函数: <code>tolower()</code>	69
3.15	将字符转换为大写英文字母函数: <code>toupper()</code>	70
第 4 章	字符串函数库: <code>string.h</code>	72
4.1	比较字符串是否相等函数: <code>bcmp()</code>	72
4.2	复制字符串函数: <code>bcopy()</code>	73
4.3	将字符串指定字节置零函数: <code>bzero()</code>	74
4.4	把内存区域的字节设置成字符函数: <code>setmem()</code>	75
4.5	从源字符中移动字节到目标字符函数: <code>movmem()</code>	76
4.6	把字符串复制到数组函数: <code>stpcpy()</code>	76

4.7	复制字符串到数组函数: strcpy()	77
4.8	字符串追加函数: strcat()	78
4.9	查找字符串首次出现位置函数: strchr()	79
4.10	字符串比较函数: strcmp()	79
4.11	字符串比较(不区分大小写)函数: stricmp()	80
4.12	字符串比较(不区分大小写)函数: strcmpi()	81
4.13	字符串查找函数: strcspn()	82
4.14	字符串复制函数: strdup()	83
4.15	字符串长度函数: strlen()	84
4.16	将字符串转换为小写形式函数: strlwr()	84
4.17	将字符串转换为大写形式函数:strupr()	85
4.18	字符串尾部追加函数: strncat()	86
4.19	字符串比较函数: strncmp()	86
4.20	字符串比较(不区分大小写)函数: strnicmp()	87
4.21	字符串比较(不区分大小写)函数: strncmpi()	88
4.22	将字符串复制到数组函数: strncpy()	89
4.23	字符串查找函数: strpbrk()	90
4.24	字符串倒序函数: strrev()	91
4.25	将字符串设置成指定字符函数: strset()	91
4.26	在字符串中查找指定字符首次出现位置函数: strstr()	92
4.27	用分隔符分解字符串函数: strtok()	93
第 5 章	标准库函数库: system.h	94
5.1	清屏函数: clrscr()	94
5.2	以指定模式刷新屏幕函数: UpdateLCD()	95
5.3	在屏幕上指定位置输出字符串函数: TextOut()	96
5.4	响铃函数: bell()	96
5.5	在屏幕上画一矩形并填充函数: block()	97
5.6	设定光标形态函数: cursor()	98
5.7	短暂延时函数: delay()	99
5.8	取汉字的点阵函数: get_chi_font()	99
5.9	取英文字符的点阵函数: get_eng_font()	101
5.10	读键函数: getkey()	102
5.11	将光标移动到指定位置函数: move()	103

5.12	调用系统例程函数: <code>noidle()</code>	103
5.13	查询汉字拼音: <code>pyfc()</code>	104
5.14	写汉字点阵函数: <code>write_chi_font()</code>	105
5.15	写英文字符的点阵函数: <code>write_eng_font()</code>	106
5.16	显示七段数码管数字函数: <code>DispBCD()</code>	107
5.17	显示滚动条函数: <code>SetScrollBar()</code>	108
第6章 图形处理函数库: <code>graphics.h</code>		109
6.1	取得最后一次调用画弧线坐标函数: <code>getarccoords()</code>	109
6.2	画弧线函数: <code>arc()</code>	110
6.3	画圆函数: <code>circle()</code>	111
6.4	绘制扇形并填充函数: <code>pieslice()</code>	112
6.5	显示当前图形模式的纵横比函数: <code>getaspectratio()</code>	113
6.6	设置图形纵横比函数: <code>setaspectratio()</code>	114
6.7	绘制并填充椭圆扇区函数: <code>sector()</code>	115
6.8	绘制椭圆函数: <code>ellipse()</code>	116
6.9	绘制并填充椭圆函数: <code>fillellipse()</code>	117
6.10	获取填充模式和填充颜色函数: <code>getfillsettings()</code>	118
6.11	将用户定义的填充模式复制到内存函数: <code>getfillpattern()</code>	120
6.12	选择用户定义的填充模式函数: <code>setfillpattern()</code>	121
6.13	设置填充模式和颜色函数: <code>setfillstyle()</code>	122
6.14	输出当前的文本属性函数: <code>settextstyle()</code>	123
6.15	获取当前图形文本字体的信息函数: <code>gettextsettings()</code>	124
6.16	在视区显示字符串函数: <code>outtext()</code>	126
6.17	在指定位置显示字符串函数: <code>outtextxy()</code>	127
6.18	注册链入的字体代码函数: <code>registerbgiFont()</code>	128
6.19	设置文本的对齐方式函数: <code>settextjustify()</code>	130
6.20	为矢量字体改变字符宽度和高度函数: <code>setusercharsize()</code>	131
6.21	获取以像素为单位的字符串高度函数: <code>textheight()</code>	132
6.22	获取以像素为单位的字符串宽度函数: <code>textwidth()</code>	133
6.23	返回最后一次不成功的图形操作的错误代码函数:	

graphresult()	135
6.24 初始化图形系统函数: initgraph()	135
6.25 获取当前视口的信息函数: getviewsettings()	136
6.26 清除图形视区函数: clearviewport()	137
6.27 获取当前图形位置的 x 坐标函数: getx()	138
6.28 获取当前图形位置的 y 坐标函数: gety()	139
6.29 为图形输出设置当前视口函数: setviewport()	140
6.30 清除图形屏幕函数: cleardevice()	141
6.31 返回屏幕的最大 x 坐标函数: getmaxx()	142
6.32 返回屏幕的最大 y 坐标函数: getmaxy()	143
6.33 将当前点移到(x, y)函数: moveto()	144
6.34 将当前点移动一相对距离函数: moverel()	145
6.35 绘制二维条形图函数: bar()	146
6.36 绘制三维条形图函数: bar3d()	147
6.37 绘制矩形函数: rectangle()	149
6.38 设置当前画线颜色函数: setcolor()	150
6.39 获取当前画线颜色函数: getcolor()	151
6.40 获取当前线型、模式和宽度函数: getlinesettings()	152
6.41 获取当前背景颜色函数: getbkcolor()	153
6.42 设置当前背景颜色函数: setbkcolor()	154
6.43 获取可以传给函数 setcolor 的最大颜色函数: getmaxcolor()	155
6.44 获取有关当前调色板的信息函数: getpalette()	156
6.45 按指定方式改变所有的调色板颜色函数: setallpalette()	158
6.46 设置有关当前调色板的信息函数: setpalette()	159
6.47 关闭图形系统函数: closegraph()	161
6.48 改变内部图形缓冲区的大小函数: setgraphbufsize()	162
6.49 通过检测硬件确定图形驱动程序和模式函数: detectgraph()	163
6.50 绘制多边形函数: drawpoly()	164
6.51 绘制多边形并填充函数: fillpoly()	165
6.52 填充有界区域函数: floodfill()	167

6.53	获取当前图形模式函数: <code>getgraphmode()</code>	168
6.54	设置当前图形模式函数: <code>setgraphmode()</code>	169
6.55	将指定区域的位图调入内存函数: <code>getimage()</code>	170
6.56	在屏幕上输出位图函数: <code>putimage()</code>	172
6.57	获取给定图形驱动程序的模式范围函数: <code>getmoderange()</code>	174
6.58	获取指定像素的颜色函数: <code>getpixel()</code>	175
6.59	在指定位置绘制像素函数: <code>putpixel()</code>	176
6.60	将所有图形设置复位为它们的默认值函数: <code>graphdefaults()</code>	178
6.61	返回一个错误信息串的指针函数: <code>grapherrormsg()</code>	179
6.62	用户可修改的图形存储区释放函数: <code>_graphfreemem()</code>	180
6.63	用户可修改的图形存储区分配函数: <code>_graphgetmem()</code>	182
6.64	返回保存位图像所需的字节数函数: <code>imagesize()</code>	183
6.65	在指定两点间画直线函数: <code>line()</code>	185
6.66	在指定两点间画直线函数: <code>lineto()</code>	186
6.67	从当前位置点到当前点绘制直线函数: <code>linereel()</code>	187
6.68	选择低亮度字符函数: <code>lowvideo()</code>	188
6.69	登录已连接进来的图形驱动程序代码函数: <code>registerbgidriver()</code>	188
6.70	将屏幕模式恢复为先前的 <code>initgraph</code> 设置函数: <code>restorecrtmode()</code>	189
6.71	设置图形输出活动页函数: <code>setactivepage()</code>	191
6.72	设置可见图形页号函数: <code>setvisualpage()</code>	192
6.73	设置当前画线宽度和类型函数: <code>setlinestyle()</code>	193
第 7 章 动态内存管理函数库: <code>alloc.h</code>		195
7.1	更改数据段空间分配函数: <code>brk()</code>	195
7.2	获取未使用内存大小函数: <code>coreleft()</code>	196
7.3	申请堆栈空间函数: <code>farcalloc()</code>	197
7.4	获取空余存储区空间大小函数: <code>farcoreleft()</code>	198
7.5	释放堆中空间函数: <code>farfree()</code>	199

7.6	存储块分配空间函数: farmalloc()	199
7.7	存储块空间调整函数: farrealloc()	200
7.8	释放已分配DOS内存块函数: freemem()	201
7.9	改变数据段空间位置函数: sbrk()	202
第8章	目录操作函数库: dir.h	204
8.1	更改工作目录函数: chdir()	204
8.2	搜索磁盘目录函数: findfirst()	205
8.3	搜索磁盘目录函数: findnext()	206
8.4	指定当前目录函数: getcurdir()	207
8.5	获取当前工作目录函数: getcwd()	208
8.6	获取磁盘驱动器号函数: getdisk()	209
8.7	建立目录函数: mkdir()	209
8.8	建立文件名函数: mktemp()	210
8.9	删除文件目录函数: rmdir()	211
8.10	设置当前工作驱动器函数: setdisk()	212
8.11	查找文件函数: searchpath()	213
8.12	构造文件名函数: fnmerge()	214
第9章	系统接口函数库: dos.h	215
9.1	磁盘读数据函数: absread()	215
9.2	磁盘写数据函数: abswrite()	216
9.3	DOS分配存储段函数: allocmem()	217
9.4	DOS系统调用函数: bdos()	218
9.5	返回国家相关信息函数: country()	218
9.6	设置Ctrl_Break处理程序函数: ctrlbrk()	219
9.7	获取扩展DOS错误信息函数: dosexterr()	220
9.8	获取Control_break设置函数: getcbrk()	221
9.9	获取DOS日期函数: getdate()	222
9.10	设置DOS日期函数: setdate()	222
9.11	设置DOS时间函数: settime()	223
9.12	从硬件端口中输入函数: inport()	224
9.13	从硬件端口中输入函数: inportb()	225
9.14	从硬件端口中输出函数: outport()	226
9.15	从硬件端口中输出函数: outportb()	226

9.16	通用DOS接口函数: <code>intdos()</code>	227
9.17	通用DOS中断接口函数: <code>intdosx()</code>	228
9.18	通用 8086 软中断接口函数: <code>int86()</code>	229
9.19	通用 8086 软中断接口函数: <code>int86x()</code>	230
9.20	改变软中断接口函数: <code>intr()</code>	231
9.21	退出并继续驻留函数: <code>keep()</code>	232
9.22	分析文件名函数: <code>parsfnm()</code>	233
9.23	检查存储单元函数: <code>peek()</code>	234
9.24	检查存储单元函数: <code>peekb()</code>	235
9.25	存值到给定存储单元函数: <code>poke()</code>	236
9.26	存值到给定存储单元函数: <code>pokeb()</code>	237
9.27	随机块读函数: <code>randbrd()</code>	237
9.28	随机块写函数: <code>randbwr()</code>	239
9.29	读段寄存器值函数: <code>segread()</code>	240
9.30	执行挂起函数: <code>sleep()</code>	241
9.31	修改DOS分配函数: <code>setblock()</code>	242
9.32	UNIX时间格式转换函数: <code>dostounix()</code>	243
9.33	获取系统时间函数: <code>gettime()</code>	244
第 10 章	输入输出函数库: <code>io.h</code>	245
10.1	打开文件函数: <code>open()</code>	245
10.2	关闭文件函数: <code>close()</code>	246
10.3	创建文件函数: <code>creat()</code>	247
10.4	文件检测结束函数: <code>eof()</code>	247
10.5	读文件函数: <code>read()</code>	248
10.6	写文件函数: <code>write()</code>	249
10.7	文件字节数统计函数: <code>filelength()</code>	251
10.8	获取文件指针位置函数: <code>tell()</code>	251
10.9	移动文件指针函数: <code>lseek()</code>	252
10.10	获取文件时间函数: <code>getftime()</code>	253
10.11	改变文件访问方式函数: <code>chmod()</code>	254
10.12	设置文件打开方式函数: <code>setmode()</code>	255
10.13	复制文件句柄函数: <code>dup()</code>	256
10.14	设备类型检查函数: <code>isatty()</code>	257

10.15	文件共享锁设置函数: lock()	258
10.16	连接文件句柄函数: fdopen()	259
10.17	文件访问权限设置函数: access()	260
第 11 章	浮点数据处理库: float.h	261
11.1	重置浮点运算系统函数: _fpreset()	261
11.2	获取浮点处理器状态值函数: _status87()	262
第 12 章	控制台输入输出函数库: conio.h	264
12.1	获取字符函数: getch()	264
12.2	清除字符函数: clrEOF()	265
12.3	清除文本函数: clrscr()	265
12.4	写字符函数: cputs()	266
12.5	删除行函数: delline()	267
12.6	文本复制函数: gettext()	268
12.7	设置光标函数: gotoxy()	269
12.8	高亮显示文本函数: highvideo()	269
12.9	插入空行函数: insline()	270
12.10	文本复制函数: puttext()	271
12.11	设置文本属性函数: textattr()	272
12.12	文本背景色选择函数: textbackground()	273
12.13	文本字符颜色选择函数: textcolor()	274
12.14	文本模式设置函数: textmode()	274
12.15	返回水平光标位置函数: whereX()	275
12.16	返回垂直光标位置函数: whereY()	275
12.17	定义活动文本窗口函数: window()	276
第 13 章	DEBUG相关函数库: assert.h	278
第 14 章	BIOS相关函数库: bios.h	279
14.1	串行IO通信函数: bioscom()	279
14.2	软硬盘IO函数: biosdisk()	280
14.3	检查设备函数: biosequip()	281
14.4	键盘接口函数: bioskey()	282
14.5	获取存储块大小函数: biosmemory()	283
14.6	设置BIOS时间函数: biostime()	283
第 15 章	内存相关函数库: mem.h	285