

最新Microsoft QuickC

2.50版

使用大全



(共四册)

周杉 王忠 编译
晓夏 武仁

■ 库函数集锦

中国科学院希望高级电脑技术公司

最新Microsoft QuickC 使 用 大 全 (V2.50版)

- 安装和运行·成套工具 (之一)
- 语言参考手册 (之二)
- 库函数集锦 (之三)
- 库函数集锦 (之四)

版权所有

翻印必究

- 北京市新闻出版局
准印证号：3202—90202
- 订购单位：北京8721信箱
- 电话：256.2329
- 地址：海淀影剧院北侧
- 乘车：320、332路至海淀黄庄站下车

引 言

《Microsoft QuickC(2.5 版)使用大全》的第三、四册全面系统地介绍了包含在 2.5 版软件包中的各个库函数,按字母顺序列出了它们的使用法、功能、原型所在头文件、返回信息、相关函数等等内容,并提供了经过精心编制的多个示例程序,读者在分析示例的过程中,可以学到大量的编程知识和技巧,从而提高自己的程序设计水平。无论对于 QuickC 的初学者还是高级程序员,这两本书都是必不可少的工具书!

由于第三、第四册主要介绍库函数的内容,所以它们的页码连续编号,书中所有函数均按字母顺序排列。

第四册的最后还列出了 QuickC2.5 中的全局变量和内容说明。

祝您成功!

目 录

库函数参考

abort	1	_bmsize	35
abs	3	bsearch	39
access	4	cabs	42
acos	6	cabsl	43
acosl	6	calloc	43
alloca	9	ceil	44
_arc	9	ceilf	46
asctime	11	_cexit	46
asin	11	_c_exit	46
assert	11	cgets	46
atan, atan2	13	_chain_intr	47
atof	14	chdir	53
atoi	16	_chdrive	55
atol, atold	16	chmod	57
_bcalloc	16	chsize	57
bdos	18	_clear87	60
Bessel	19	clearerr	62
_bexpand	20	_clearscreen	63
_freseg	21	clock	65
_bheapadd	23	close	65
_bheapchk	23	_control87	68
_bheapmin	23	cos	69
_bheapseg	23	cosh	71
_bheapset	24	coshf	71
_bheapwalk	24	cosl	72
_bios_disk	25	cprintf	72
_bios_equiplist	27	cputs	73
_bios_keybrd	29	creat	75
_bios_memsiz	31	escanf	76
_bios_serialcom	32	ctime	77
_bios_timeofday	33	diecetomsbin	80
_bmalloc	35	difftime	81
		_disable	83

<code>_displaycursor</code>	83	<code>execle</code>	119
<code>div</code>	85	<code>execlp</code>	119
<code>dmsbintoieee</code>	85	<code>execlpe</code>	119
<code>_dos_allocmem</code>	85	<code>execv</code>	120
<code>_dos_close</code>	87	<code>execve</code>	121
<code>_dos_creat</code>	89	<code>execvp</code>	121
<code>_dos_creatnew</code>	90	<code>execvpe</code>	122
<code>dosexterr</code>	90	<code>exit</code>	122
<code>_dos_findfirst</code>	92	<code>exp</code>	122
<code>_dos_findnext</code>	95	<code>_expand</code>	124
<code>_dos_freemem</code>	95	<code>expl</code>	125
<code>_dos_getdate</code>	96	<code>fabs</code>	125
<code>_dos_getdiskfree</code>	98	<code>fclose</code>	126
<code>_dos_getdrive</code>	98	<code>fcloseall</code>	128
<code>_dos_getfileattr</code>	99	<code>fcvt</code>	128
<code>_dos_getftime</code>	101	<code>fdopen</code>	128
<code>_dos_gettime</code>	102	<code>feof</code>	130
<code>_dos_getvect</code>	102	<code>ferror</code>	131
<code>_dos_keep</code>	104	<code>fexpand</code>	131
<code>_dos_open</code>	105	<code>fflush</code>	131
<code>_dos_read</code>	105	<code>_ffree</code>	133
<code>_dos_setblock</code>	106	<code>fgetc,fgetchar</code>	134
<code>_dos_setdrive</code>	106	<code>fgetchar</code>	136
<code>_dos_setfileattr</code>	106	<code>fgetpos</code>	136
<code>_dos_setftime</code>	107	<code>fgets</code>	139
<code>_dos_settime</code>	107	<code>_fheapchk</code>	139
<code>_dos_setvect</code>	108	<code>_fheapmin</code>	139
<code>_dos_write</code>	108	<code>_fheapset</code>	141
<code>dup</code>	109	<code>_fheapwalk</code>	142
<code>dup2</code>	110	<code>fieetomsbin</code>	142
<code>ecvt</code>	110	<code>filelength</code>	142
<code>_ellipse</code>	112	<code>fileno</code>	144
<code>_emit</code>	114	<code>_floodfill</code>	146
<code>_enable</code>	115	<code>floor</code>	148
<code>eof</code>	115	<code>floorl</code>	148
<code>execl</code>	117	<code>flushall</code>	148

_fmemccpy	149	_fstrlen	181
_fmemchr	151	_fstrlwr	181
_fmemcmp	153	_fstrncat	182
_fmemcpy	155	_fstrncmp	184
_fmemicmp	155	fstrncpy	184
_fmemmove	155	_fstrnicmp	185
_fmemset	156	_fstrnset	185
fmod	157	_fstrpbrk	185
fmodl	157	_fstrchr	186
fmsbintoieee	157	_fstrrev	186
_fmize	158	_fstrset	186
fopen	158	fstrspn	186
FP_OFF	158	_fstrstr	187
_fpreset	160	_fstrtok	187
fprintf	163	_fstrlwr	187
fputc	165	ftell	187
fputs	165	ftime	188
fread	165	_fullpath	189
_frealloc	168	fwrite	190
free	168	gevt	190
freect	168	getarcinfo	190
freopen	170	_getbkcolor	191
frexp	170	getc	193
frexpl	171	getch	195
fscanf	171	getchar	196
fseek	171	getche	196
fsetpos	172	_getcolor	197
fsopen	172	_getcurrentposition	201
fstat	175	getcwd	201
_fstreat	177	_getcwd	202
fstrchr	178	_getdrive	202
fstrcmp	178	getenv	202
fstrcpy	178	_getfillmask	204
_fstrcspn	179	_getfontinfo	204
_fstrdup	180	_gettexttent	207
fstricmp	180	_getgtextvector	207

<code>_getimage</code>	208
<code>_getlinestyle</code>	210
<code>_getphyscoord</code>	210
<code>_getpid</code>	210
<code>_getpixel</code>	211
<code>gets</code>	212
<code>_gettextcolor</code>	213
<code>_gettextcursor</code>	214
<code>_gettextposition</code>	214
<code>_getvideoconfig</code>	214
<code>_getviewcoord</code>	218
<code>_getvisualpage</code>	218
<code>getw</code>	218
<code>_getwindowcoord</code>	219
<code>_getwritemode</code>	219
<code>gmtime</code>	221
<code>_grstaus</code>	222
<code>halloc</code>	225
<code>_harderr</code>	226
<code>_hardresume</code>	231
<code>_heapadd</code>	231
<code>_heapchk</code>	232
<code>_heapmin</code>	232
<code>_heapset</code>	233
<code>_heapwalk</code>	234
<code>hfree</code>	234
<code>hypot</code>	235
<code>hypotl</code>	236
<code>_imagesize</code>	236
<code>inpw</code>	237
<code>inpw</code>	239
<code>int86</code>	240
<code>int86x</code>	240
<code>intdos</code>	241
<code>intdosx</code>	241
<code>isalnum</code>	242

<code>isatty</code>	244
<code>itoa</code>	244
<code>j0</code>	246
<code>_j0l</code>	247
<code>kbhit</code>	247
<code>labs</code>	248
<code>ldexp</code>	248
<code>ldexpl</code>	249
<code>ldiv</code>	249
<code>lfind</code>	249
<code>_lineto</code>	250
<code>localconv</code>	252
<code>localtime</code>	252
<code>locking</code>	253
<code>log</code>	256
<code>_makepath</code>	256
<code>malloc</code>	257
<code>_matherr</code>	259
<code>_matherrl</code>	261
<code>max</code>	261
<code>_memavl</code>	262
<code>memccpy</code>	262
<code>memchr</code>	262
<code>memcmp</code>	263
<code>memcpy</code>	263
<code>memicmp</code>	264
<code>_memmax</code>	264
<code>memmove</code>	265
<code>memset</code>	265
<code>max,min</code>	266
<code>mkdir</code>	266
<code>mktemp</code>	266
<code>mktime</code>	267
<code>modf</code>	268
<code>modfl</code>	268
<code>movedata</code>	268

<code>_moveto</code>	269	<code>_pg_initchart</code>	293
<code>_msize</code>	269	<code>_pg_resetpalette</code>	294
<code>_ncalloc</code>	270	<code>_pg_resetstyleset</code>	294
<code>_nexpand</code>	270	<code>_pg_setchardef</code>	294
<code>_nfree</code>	270	<code>_pg_setpalette</code>	295
<code>_nheapchk</code>	271	<code>_pg_setstyleset</code>	295
<code>_nheapmin</code>	271	<code>_pg_vlabelchart</code>	295
<code>_nheapset</code>	271	<code>_pie</code>	296
<code>_nheapwalk</code>	272	<code>_polygon</code>	296
<code>_nmalloc</code>	272	<code>pow</code>	297
<code>_nrealloc</code>	273	<code>powl</code>	298
<code>_nstrdup</code>	273	<code>printf</code>	298
<code>offset</code>	273	<code>putc</code>	299
<code>onexit</code>	274	<code>putch</code>	300
<code>open</code>	274	<code>putchar</code>	301
<code>outgtxt</code>	275	<code>putenv</code>	302
<code>_outmem</code>	275	<code>_putimage</code>	302
<code>outp</code>	276	<code>puts</code>	303
<code>outpw</code>	276	<code>putw</code>	304
<code>_outtext</code>	277	<code>qsort</code>	305
<code>perror</code>	277	<code>raise</code>	305
<code>_pg_analyzechart</code>	278	<code>rand</code>	306
<code>_pg_analyzechartms</code>	280	<code>read</code>	306
<code>_pg_analyzepie</code>	280	<code>realloc</code>	307
<code>_pg_analyzescatter</code>	281	<code>_rectangle</code>	308
<code>_pg_analyzescatterms</code>	281	<code>_registerfonts</code>	309
<code>_pg_chart</code>	282	<code>_remapallpalette</code>	309
<code>_pg_chartms</code>	283	<code>remapalette</code>	313
<code>_pg_chartpie</code>	286	<code>remove</code>	313
<code>_pg_chartscatter</code>	286	<code>rename</code>	314
<code>_pg_chartscatterms</code>	288	<code>rewind</code>	314
<code>_pg_defaultchart</code>	288	<code>rmdir</code>	315
<code>_pg_getchardef</code>	289	<code>rmtmp</code>	315
<code>_pg_getpalette</code>	290	<code>_rotl</code>	315
<code>_pg_getstyleset</code>	293	<code>_rotr</code>	317
<code>_pg_hlabelchart</code>	293	<code>scanf</code>	317

<code>_scrolltextwindow</code>	318
<code>_searchdev</code>	321
<code>segread</code>	321
<code>_selectpalette</code>	321
<code>_setactivepage</code>	323
<code>_setbkcolor</code>	326
<code>setbuf</code>	326
<code>_setcliprgn</code>	328
<code>_setcolor</code>	331
<code>_setfillmask</code>	331
<code>_setfont</code>	331
<code>_setgtextvector</code>	333
<code>setjmp</code>	334
<code>_setlinestyle</code>	334
<code>setlocale</code>	334
<code>setmode</code>	335
<code>_setpixel</code>	337
<code>_setttextcolor</code>	338
<code>_setttextcursor</code>	339
<code>_setttextposition</code>	339
<code>_setttextrows</code>	340
<code>_setttextwindow</code>	340
<code>setvbuf</code>	341
<code>_setvideomode</code>	341
<code>_setvideomoderows</code>	346
<code>_setvieworg</code>	346
<code>_setviewport</code>	347
<code>_setvisualpage</code>	347
<code>_setwindow</code>	347
<code>_setwritemode</code>	348
<code>signal</code>	348
<code>sin</code>	350
<code>sinh</code>	351
<code>sinhl</code>	351
<code>sinl</code>	351
<code>sopen</code>	352

<code>spawnl</code>	353
<code>spawnv</code>	357
<code>_splitpath</code>	358
<code>sprint</code>	358
<code>sqr</code>	359
<code>srand</code>	359
<code>sscanf</code>	359
<code>stackavail</code>	360
<code>stat</code>	360
<code>_status87</code>	360
<code>stdin,stdout,stderr,stderr,</code>	
<code>stdprn</code>	360
<code>strcat</code>	361
<code>strchr</code>	363
<code>strcmp</code>	363
<code>stricmp</code>	364
<code>strcoll</code>	365
<code>strcpy</code>	365
<code>strcspn</code>	366
<code>_strdate</code>	366
<code>strdup</code>	367
<code>strerror</code>	367
<code>_strerror</code>	368
<code>strtime</code>	368
<code>stricmp</code>	370
<code>ssrten</code>	370
<code>strlwr</code>	370
<code>strncat</code>	371
<code>strncmp</code>	372
<code>strncpy</code>	372
<code>strnicmp</code>	372
<code>strnset</code>	373
<code>strpbrk</code>	373
<code>strchr</code>	374
<code>strrev</code>	374
<code>strset</code>	374

strspn	375
strstr	375
_strtime	375
strtod	375
strtok	378
strtol	378
strupr	379
strxfrm	379
swab	379
system	380
tan	381
tanh	381
tanhf	381
tanl	381
tell	381
time	381
tmpfile	382
tmpnam	382
toascii	384
tzset	384
ultoi	385
umask	385

ungetc	387
ungetch	388
unlink	389
_unregisterfont	389
utime	390
va...	391
vfprintf	394
_wapon	396
write	397
y0,y1,yn	398
_y0l,_y1l,ynl	398

全局变量

_amblksize	399
daylight,timezone,tzname	399
_doserrno,errno,sys_errlist,	
sys_nerr	400
environ	401
_fmode	401
_osmajor,_osminor,_osmode,	
_osversion	401
_psp	402

_getcurrentposition

▲包含在: <graph.h>

▲用法: struct xycoord _getcurrentposition(void);
struct _wxycoord _getcurrentposition_w(void);

▲返回: 以 xycoord 结构返回当前位置坐标。

▲参见: _grstatus, _lineto, _moveto, _outtext

▲说明:

_getcurrentposition 返回当前图形输出位置的坐标值。返回值以 xycoord 结构形式存放(该结构在 GRAPH.H 中定义), _getcurrentposition_w 以 _wxycoord 结构返回当前位置(该结构在 GRAPH.H 中定义)。

当前位置可由 _Lineto, _Moveto, 和 outtext 改变。

使用 _setvideomode, setvideomoderows, _setviewport 后位置的缺省值为视口中心。

这些函数只影响图形输出, 不影响文本输出。文本输出在当前文本位置开始(见

_settextposion)

▲返回值:

返回当前图形的坐标值, 无错误返回。

▲示例: 参见 _getcolor()。

getcwd

▲包含在: <direct.h>, <errno.h>

▲用法: char *getcwd(char *buffer, int maxlen);
char *_getdcwd(int drive, char *buffer, int maxlen);

▲返回: 如果成功, 返回指向 buffer 的指针。否则, 返回 NULL, 并置 errno 为: ENOMEM, ERANGE。

▲参见: chdir, _getdrive, mkdir, rmdir

▲说明:

getcwd 取当前工作目录路径名, 并将其存入 buffer, 用 _getdcwd, 则路径名中包括所使用的驱动器, (0 表示缺省驱动器, 1 = A, 2 = B 等)。

整型参数 maxlen 说明了路径名的最大长度, 若路径名长度(包括结尾的空字符)超出了 maxlen, 则会出现错误, STDLIB.H 中的 _MAX_PATH 常量定义了最大的路径长度, <buffer> 参数也可以为 NULL, 用来存放路径名的长度至少为 maxlen 的缓冲区将自动被分配(用 malloc)最后可用 free 函数(参数为被分配的缓冲区指针)释放。

▲返回值:

_getcwd 和 _getdcwd 返回 path, 若返回值为 NULL, 则表示出错, 并置 errno 为 ENOMEM 或 ERANGE。

▲示例: 参见 _chdrive()。

_getdcwd

- ▲包含在: <direct.h>, <errno.h>
- ▲用法: `char *getcwd( char *buffer, int maxlen );`
`char *_getdcwd( int drive, char *buffer, int maxlen );`
- ▲返回: 如果成功, 返回指向 `buffer` 的指针。否则, 返回 `NULL`。并置 `errno` 为: `ENOMEM`, `ERANGE`。
- ▲参见: `chdir`, `_getdrive`, `mkdir`, `rmdir`
- ▲说明: 参见 `getcwd`。
- ▲示例: 参见 `getcwd`。

_getdrive

- ▲包含在: <direct.h>
- ▲用法: `int _getdrive( void );`
- ▲返回: 当前工作驱动器名(1=A, 2=B, etc)。
- ▲参见: `_chdrive`, `_dos_getdrive`, `_dos_setdrive`, `_getcwd`, `_getdcwd`
- ▲说明:
`_getdrive` 返回当前工作驱动器名(1 = A, 2 = B 等)
- ▲返回值:
- ▲示例: 参见 `_chdrive()`。

getenv

- ▲包含在: <stdlib.h>
- ▲用法: `char *getenv( char *varname );`
- ▲返回: 返回指向 `varname` 在环境表中所占项的指针。如该变量尚未定义, 则返回 `NULL`。
- ▲参见: `putenv`
- ▲说明:

`getenv` 查询环境变量表, 以找出与 `varname` 相对应的项。

环境变量定义了在其中运行进程的一个环境(例如, 用于联结的程序库的缺省搜索路径), `getenv` 函数的使用须区分大小写。

环境表中的项不能直接改变, 如要改变一个项, 应使用 `putenv` 函数, 为了修改返回值, 而不致影响环境表, 可使用 `strdup` 或 `strcpy` 来产生该符号串的一个拷贝。`getenv` 和 `putenv` 用全局变量 `environ` 来访问环境表, `putenv` 可能改变 `environ` 的值因此使得参数 `envp` 在 `main` 函数中失效。所以使用 `environ` 变量来访问环境表是安全的。

- ▲返回值:
`getenv` 函数返回指向环境变量表中包含 `varname` 项的指针。如该变量还未定义, 则返回 `NULL`。

▲示 例:

/* ENVIRON.C 演示了下列环境变量函数的使用:

```
    getenv, putenv, _searchenv.  /*
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

void main()
{
    char *pathvar, pathbuf[128], filebuf[128];

    /* Get the PATH environment variable and save a copy of it. */
    pathvar = getenv( "PATH" );
    strcpy( pathbuf, pathvar );
    printf( "Old PATH: %s\n", pathvar ? pathvar : "variable not set");

    /* Add a new directory to the path. */
    strcat( pathbuf, ";\\QC;" );
    if( putenv( pathbuf ) == -1 )
    {
        printf( "Failed\n");
        exit( 1 );
    }
    else
        printf( "New PATH: %s\n", pathbuf );

    /* Search for file in the new path. */
    _searchenv( "QC.INI", "PATH", filebuf );
    if( *filebuf )
        printf( "QC.INI found at %s\n", filebuf );
    else
        printf( "QC.INI not found\n" );

    /* Restore original path. */
    if( putenv( pathvar ) == -1 )
        printf( "Failed\n");
    else
        printf( "Old PATH: %s\n", pathvar );
    exit( 0 );
}
```

_getfillmask

▲包含在: <graph.h>

▲用法: unsigned char _far *_getfillmask(unsigned char _far *mask);

▲返回: mask 中的当前填充掩码, 如果没有设置, 则返回 NULL。

▲参见: _ellipse, _floodfill, _pie, _polygon, _rectangle,
_setfillmask

▲说明:

_getfillmask 返回当前填充掩码, 该掩码为一个 8×8 位数组, 其中每一位代表一个像素位, 某位为 1 则相应的像素为当前颜色, 为 0 则相应的像素颜色不变。

掩码在整个填充域上重复如没有设置填充掩码或为 NULL 则在填充中仅用当前颜色。

▲返回值:

没有设置掩码时, 该函数返回 UNLL。

▲示例: 参见 _floodfill。

_getfontinfo

▲包含在: <graph.h>

▲用法: short _getfontinfo(struct _fontinfo _far *fontbuffer);

▲返回: 如果没有字型被注册或装入, 则返回一个负数。

▲参见: _gettextextent, _outgtext, _registerfonts, _setfont,
_setgtextvector, _unregisterfonts

▲说明:

_getfontinfo 取得当前字形特征并存于 _fontinfo 结构(在 GRAPH.H 中定义)变量中。

▲返回值:

若某一字型没被装入或登记, 则返回负值, 否则返回非负值。

▲示例:

/* FONTS.C 演示了下列函数的使用

```
_registerfonts      _setfont      _outgtext
_unregisterfonts    _getfontinfo    _getgtextent
_setgtextvector
```

*/

```
#include <conio.h>
```

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

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

```
#include <string.h>
```

```
#include <graph.h>
```

```
#define NFonts 6
```

```
unsigned char *face[NFonts] =
```

```
{
```

```
    "Courier", "Helvetica", "Times Roman", "Modern", "Script", "Roman"
```

```
};
```

```
unsigned char *options[NFonts] =
```

```
{
```

```
    "courier", "helv", "tms rma", "modera", "script", "roman"
```

```
};
```

```
void main()
```

```
{
```

```
    unsigned char list[20];
```

```
    char fndir[_MAX_PATH];
```

```
    struct videoconfig vc;
```

```
    struct _fontinfo fi;
```

```
    short nfont, x, y;
```

```
    /* Read header info from .FON files in current or given directory. */
```

```
    if( !_registerfonts( "*.FON" ) <= 0 )
```

```
    {
```

```
        _outtext( "Enter full path where .FON files are located: " );
```

```
        gets( fndir );
```

```
        strcat( fndir, "\\*.FON" );
```

```
        if( !_registerfonts( fndir ) <= 0 )
```

```
        {
```

```
            _outtext( "Error: can't register fonts" );
```

```
            exit( 1 );
```

```
        }
```

```
    }
```

```
    /* Set highest available graphics mode and get configuration. */
```

```
    if( !_setvideomode( _MAXRESMODE ) )
```

```
        exit( 1 );
```

```
    _getvideoconfig( &vc );
```

```
    /* Display each font name centered on screen. */
```

```
    for( nfont = 0; nfont < NFonts; nfont++ )
```

```
    {
```

```

/* Build options string. */
strcat( strcat( strcpy( list, "t" ), options[nfont] ), "" );
strcat( list, "h30w24b" );

_clearscreen( _GCLEARSCREEN );
if( _setfont( list ) >= 0 )
{
    /* Use length of text and height of font to center text. */
    x = (vc.numxpixels / 2) - (_gettexttextent( face[nfont] ) / 2);
    y = (vc.numypixels / 2) + (_gettexttextent( face[nfont] ) / 2);
    if( _getfontinfo( &fi ) )
    {
        _outtext( "Error: Can't get font information" );
        break;
    }
    _moveto( x, y );
    if( vc.numcolors > 2 )
        _setcolor( nfont + 2 );

    /* Rotate and display text. */
    _settextvector( 1, 0 );
    _outtext( face[nfont] );
    _settextvector( 0, 1 );
    _outtext( face[nfont] );
    _settextvector( -1, 0 );
    _outtext( face[nfont] );
    _settextvector( 0, -1 );
    _outtext( face[nfont] );
}
else
{
    _outtext( "Error: Can't set font: " );
    _outtext( list );
}
getch();
}
_unregisterfonts();
_setvideomode( _DEFAULTMODE );
}

```