**【C语言】控制台窗口图形界面编程（六）：光标设置**



[沧海一笑-dj](https://dengjin.blog.csdn.net/) 2019-05-25 18:45:14  1305  收藏 28

分类专栏： [C和C++](https://blog.csdn.net/dengjin20104042056/category_2700349.html) 文章标签： [CONSOLE\_CURSOR\_INFO](https://so.csdn.net/so/search/s.do?q=CONSOLE_CURSOR_INFO&t=blog&o=vip&s=&l=&f=&viparticle=) [GetConsoleCursorInfo](https://so.csdn.net/so/search/s.do?q=GetConsoleCursorInfo&t=blog&o=vip&s=&l=&f=&viparticle=) [SetConsoleCursorInfo](https://so.csdn.net/so/search/s.do?q=SetConsoleCursorInfo&t=blog&o=vip&s=&l=&f=&viparticle=) [SetConsoleCursorPosition](https://so.csdn.net/so/search/s.do?q=SetConsoleCursorPosition&t=blog&o=vip&s=&l=&f=&viparticle=) [光标控制](https://so.csdn.net/so/search/s.do?q=%E5%85%89%E6%A0%87%E6%8E%A7%E5%88%B6&t=blog&o=vip&s=&l=&f=&viparticle=)

**00. 目录**

**文章目录**

* + [00. 目录](https://blog.csdn.net/dengjin20104042056/article/details/90550212?ops_request_misc=%25257B%252522request%25255Fid%252522%25253A%252522160855382516780266267772%252522%25252C%252522scm%252522%25253A%25252220140713.130102334.pc%25255Fall.%252522%25257D&request_id=160855382516780266267772&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-7-90550212.first_rank_v2_pc_rank_v29&utm_term=CONSOLE_CURSOR_INFO#00__0)
  + [01. CONSOLE\_CURSOR\_INFO结构](https://blog.csdn.net/dengjin20104042056/article/details/90550212?ops_request_misc=%25257B%252522request%25255Fid%252522%25253A%252522160855382516780266267772%252522%25252C%252522scm%252522%25253A%25252220140713.130102334.pc%25255Fall.%252522%25257D&request_id=160855382516780266267772&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-7-90550212.first_rank_v2_pc_rank_v29&utm_term=CONSOLE_CURSOR_INFO#01_CONSOLE_CURSOR_INFO_3)
  + [02. GetConsoleCursorInfo函数](https://blog.csdn.net/dengjin20104042056/article/details/90550212?ops_request_misc=%25257B%252522request%25255Fid%252522%25253A%252522160855382516780266267772%252522%25252C%252522scm%252522%25253A%25252220140713.130102334.pc%25255Fall.%252522%25257D&request_id=160855382516780266267772&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-7-90550212.first_rank_v2_pc_rank_v29&utm_term=CONSOLE_CURSOR_INFO#02_GetConsoleCursorInfo_24)
  + [03. SetConsoleCursorInfo函数](https://blog.csdn.net/dengjin20104042056/article/details/90550212?ops_request_misc=%25257B%252522request%25255Fid%252522%25253A%252522160855382516780266267772%252522%25252C%252522scm%252522%25253A%25252220140713.130102334.pc%25255Fall.%252522%25257D&request_id=160855382516780266267772&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-7-90550212.first_rank_v2_pc_rank_v29&utm_term=CONSOLE_CURSOR_INFO#03_SetConsoleCursorInfo_92)
  + [04. SetConsoleCursorPosition函数](https://blog.csdn.net/dengjin20104042056/article/details/90550212?ops_request_misc=%25257B%252522request%25255Fid%252522%25253A%252522160855382516780266267772%252522%25252C%252522scm%252522%25253A%25252220140713.130102334.pc%25255Fall.%252522%25257D&request_id=160855382516780266267772&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-7-90550212.first_rank_v2_pc_rank_v29&utm_term=CONSOLE_CURSOR_INFO#04_SetConsoleCursorPosition_185)

**01. CONSOLE\_CURSOR\_INFO结构**

包含有关控制台光标的信息。

typedef struct \_CONSOLE\_CURSOR\_INFO {

DWORD dwSize;

BOOL bVisible;

} CONSOLE\_CURSOR\_INFO, \*PCONSOLE\_CURSOR\_INFO;

* 1
* 2
* 3
* 4

**dwSize**  
光标填充的字符单元格的百分比。该值介于1和100之间。光标外观会发生变化，从完全填充单元格到显示为单元格底部的水平线。

**注意** 虽然**dwSize**值通常介于1和100之间，但在某些情况下，可能会返回该范围之外的值。例如，如果在注册表**中将CursorSize**设置为0，则返回的**dwSize**值将为0。

**bVisible**  
光标的可见性。如果光标可见，则此成员为**TRUE**。

**02. GetConsoleCursorInfo函数**

获取有关指定控制台屏幕缓冲区的光标大小和可见性的信息。

**类型声明**

BOOL WINAPI GetConsoleCursorInfo(

\_In\_ HANDLE hConsoleOutput,

\_Out\_ PCONSOLE\_CURSOR\_INFO lpConsoleCursorInfo

);

功能:

获取光标相关信息

参数:

hConsoleOutput 控制台屏幕缓冲区的句柄。句柄必须具有GENERIC\_READ访问权限。

lpConsoleCursorInfo 指向CONSOLE\_CURSOR\_INFO结构的指针，该结构接收有关控制台游标的信息。

返回值:

如果函数成功，则返回值为非零值。

如果函数失败，则返回值为零。要获取扩展错误信息，请调用GetLastError。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14

**参考程序**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include <Windows.h>

#include <conio.h>

int main(void)

{

//定义句柄变量

HANDLE hOut = NULL;

//光标信息变量

CONSOLE\_CURSOR\_INFO cursorInfo;

//获取标准输出句柄

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

//获取当前光标信息

GetConsoleCursorInfo(hOut, &cursorInfo);

printf("光标默认大小: %d 光标是否可见: %d\n", cursorInfo.dwSize, cursorInfo.bVisible);

//关闭句柄

CloseHandle(hOut);

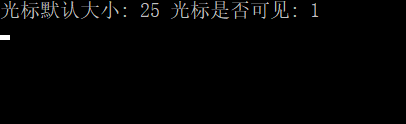
system("pause");

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32

**执行结果**



**03. SetConsoleCursorInfo函数**

设置指定控制台屏幕缓冲区的光标大小和可见性。

**函数声明:**

BOOL WINAPI SetConsoleCursorInfo(

\_In\_ HANDLE hConsoleOutput,

\_In\_ const CONSOLE\_CURSOR\_INFO \*lpConsoleCursorInfo

);

功能：

设置光标的属性

参数：

hConsoleOutput 控制台屏幕缓冲区的句柄。句柄必须具有GENERIC\_READ访问权限。

lpConsoleCursorInfo 指向CONSOLE\_CURSOR\_INFO结构的指针，该结构为控制台屏幕缓冲区的游标提供新规范。

返回值：

如果函数成功，则返回值为非零值。

如果函数失败，则返回值为零。要获取扩展错误信息，请调用GetLastError。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13

**参考程序**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include <Windows.h>

#include <conio.h>

int main(void)

{

//定义句柄变量

HANDLE hOut = NULL;

//光标信息变量

CONSOLE\_CURSOR\_INFO cursorInfo;

//获取标准输出句柄

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

//获取当前光标信息

GetConsoleCursorInfo(hOut, &cursorInfo);

printf("光标默认大小: %d 光标是否可见: %d\n", cursorInfo.dwSize, cursorInfo.bVisible);

getchar();

//设置光标的尺寸为20

cursorInfo.dwSize = 5;

SetConsoleCursorInfo(hOut, &cursorInfo);

getchar();

//设置光标的尺寸为50

cursorInfo.dwSize = 50;

SetConsoleCursorInfo(hOut, &cursorInfo);

getchar();

//设置光标的尺寸为100

cursorInfo.dwSize = 100;

SetConsoleCursorInfo(hOut, &cursorInfo);

getchar();

//设置光标不可见

cursorInfo.dwSize = 25;

cursorInfo.bVisible = 0;

SetConsoleCursorInfo(hOut, &cursorInfo);

getchar();

//关闭句柄

CloseHandle(hOut);

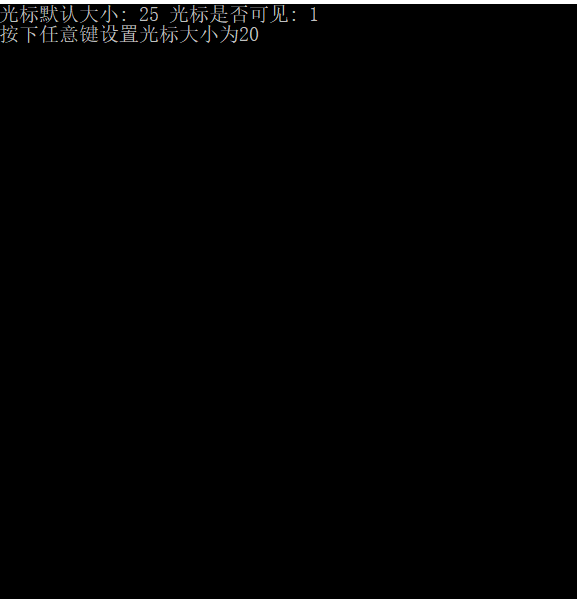
system("pause");

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56

**执行结果**



**04. SetConsoleCursorPosition函数**

设置指定控制台屏幕缓冲区中的光标位置。

**函数声明：**

BOOL WINAPI SetConsoleCursorPosition(

\_In\_ HANDLE hConsoleOutput,

\_In\_ COORD dwCursorPosition

);

功能：

设置光标的位置

参数：

hConsoleOutput 控制台屏幕缓冲区的句柄。句柄必须具有GENERIC\_READ访问权限。

dwCursorPosition 用于指定新的光标位置（以字符为单位）。坐标是屏幕缓冲区字符单元格的列和行。坐标必须位于控制台屏幕缓冲区的边界内。

返回值：

如果函数成功，则返回值为非零值。

如果函数失败，则返回值为零。要获取扩展错误信息，请调用GetLastError。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15

**参考程序；**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include <Windows.h>

#include <conio.h>

int main(void)

{

//定义句柄变量

HANDLE hOut = NULL;

COORD pos;

//获取标准输出句柄

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

pos.X = 8;

pos.Y = 8;

//设置光标的位置

SetConsoleCursorPosition(hOut, pos);

printf("hello itcast1");

pos.X = 18;

pos.Y = 16;

//设置光标的位置

SetConsoleCursorPosition(hOut, pos);

printf("hello itcast2");

//关闭句柄

CloseHandle(hOut);

system("pause");

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38

**测试结果：**  
