Colores y coordenadas en la consola de windows

Especificar posición y colores del texto.

Windows cuenta con dos funciones útiles para mejorar la apariencia de la visualización en la consola.

Una (`SetConsoleCursorPosition`) permite establecer en qué lugar de la pantalla se va a mostrar el próximo caracter que se imprima (ej. con `printf()`...) y otra función (`SetConsoleTextAttribute`) da la posibilidad de establecer color de letra y color de fondo.

Para utilizar estas funciones es necesario añadir `windows.h`

```
#include <windows.h>
```

Windows usa un identificador (que se llama "**handle**") para poder hacer referencia a cada recurso que maneja (ej. ventana, botón, puerto usb, etc.)

Para que las funciones de windows interactúen con un recurso determinado es necesario obtener su handle; entonces para usar funciones de windows con la línea de comandos hay que obtener su handle; se hace así:

```
// declaro una variable tipo HANDLE

HANDLE cmd;

// le asigno el HANDLE de línea de comandos

cmd = GetStdHandle( STD_OUTPUT_HANDLE );
```

Con esta referencia a la ventana de línea de comandos podremos interactuar con ella.

SetConsoleCursorPosition - fijando coordenadas

`SetConsoleCursorPosition( hcmd, coordenadas);` especifica que el próximo caracter que se envíe a la consola aparece en la posición indicada por coordenadas.

La posición (0, 0) es la esquina superior izquierda. Las coordenadas se indican con una variable de un tipo especial de windows: **`COORD`**

Para poder especificar coordenadas, hay que definir una variable de este tipo:

```
COORD pos;
```

Luego de definirla se puede usar así:

`pos.X` es la parte de pos que guarda la distancia desde el borde izquierdo.

`pos.Y` es la parte de pos que guarda la distancia desde el borde superior.

Por ejemplo, si para mostrar una X en la cuarta línea, a 10 lugares del margen izquierdo (Y = 3, X = 9):

```
posi.X = 9;

posi.Y = 3;

SetConsoleCursorPosition(hcmd, posi);

printf("X");
```

Recordando que hcmd es el handle, el "identificador" del que hablamos previamente y posi es la coordenada.

Un ejemplo completo que muestra una X en X=9, Y=3

```
#include <stdio.h>

#include <windows.h>
```

```
int main()
{
	COORD posi;
	HANDLE hcmd;
	hcmd = GetStdHandle( STD_OUTPUT_HANDLE );
	posi.X = 9;
	posi.Y = 3;
	SetConsoleCursorPosition(hcmd, posi);
	printf("X");
	return 0;
```

SetConsoleTextAttribute - poner color

Para establecer el color de letra y de fondo en la consola, se usa

SetConsoleTextAttribute(hcmd, atributos);

Donde **hcmd** es el handle a la ventana de línea de comandos y **atributos** es uno o más de los siguientes valores separados por | :

FOREGROUND_BLUE	**Color de texto AZUL**	**BACKGROUND_BLUE**	**Color de fondo AZUL**
FOREGROUND_GREEN	**Color de texto VERDE**	**BACKGROUND_GREEN**	**Color de fondo VERDE**
FOREGROUND_RED	**Color de texto ROJO**	**BACKGROUND_RED**	**Color de fondo ROJO**
FOREGROUND_INTENSITY	**Texto Alta intensidad.**	**BACKGROUND_INTENSITY**	**Fondo Alta intensidad**

Para frente o fondo hay 3 colores disponibles: rojo, verde, azul, que se pueden combinar. Además, el color puede ser “brillante” agregando FOREGROUND_INTENSITY o BACKGROUND_INTENSITY) según corresponda letra o fondo.

Estos tres colores y si es brillante o no da 16 posibilidades de color, tanto para texto como para fondo.

Ej.: escribir con letras color verde:

```
SetConsoleTextAttribute(hcmd, FOREGROUND_GREEN);
```

y para texto verde brillante (alta intensidad) combino "green" con "intensity" mediante el operador de bits OR (|)

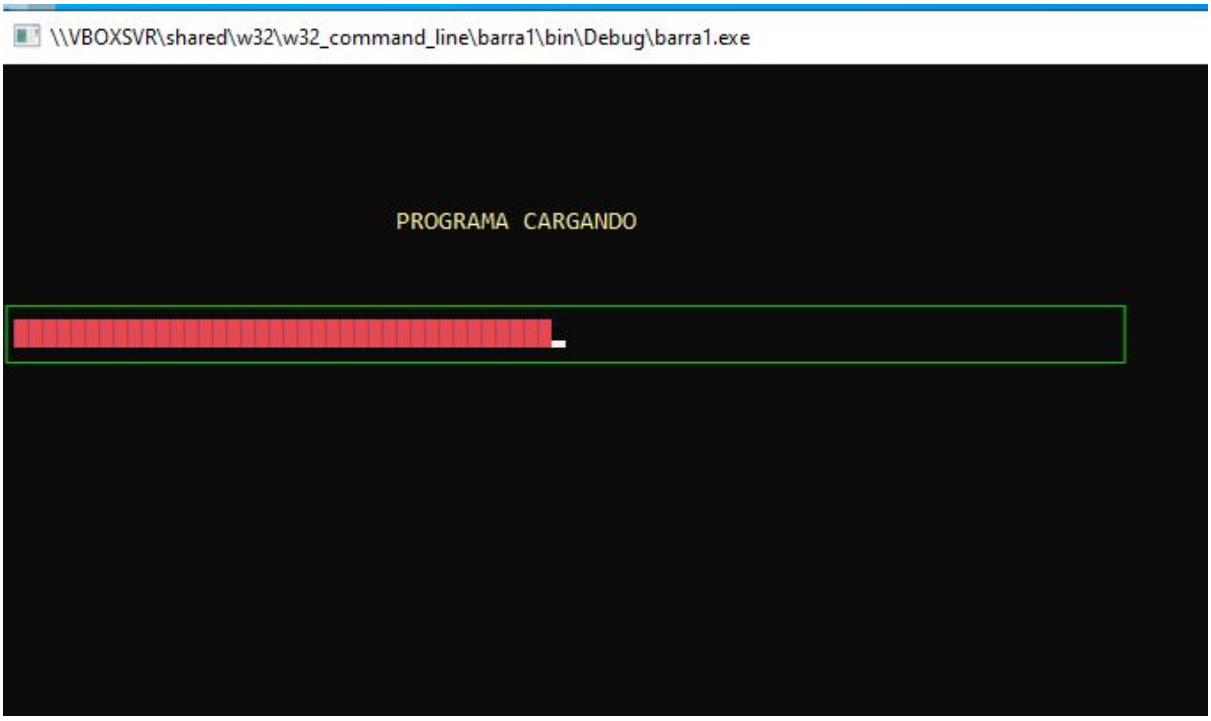
```
SetConsoleTextAttribute(hcmd,
            FOREGROUND_GREEN |
            FOREGROUND_INTENSITY
            );
```

terminal de windows mostrando los colores disponibles

El siguiente ejemplo muestra texto amarillo alta intensidad sobre fondo azul:

(rojo y verde hacen amarillo)

```
SetConsoleTextAttribute(hcmd,
            FOREGROUND_RED |
            FOREGROUND_GREEN |
            FOREGROUND_INTENSITY |
            BACKGROUND_BLUE
            );
```



una barra que muestra progreso

Otras funciones interesantes

- **`Sleep(milisegundos);`**

Bloquea el programa la cantidad de milisegundos indicada.

- **`SetConsoleTitle("Monitor de proceso");`**

Cambia nombre a la ventana de consola.

- **SetConsoleCursorInfo(h,&curInfo);**

Fija el alto del cursor y si es visible o no...

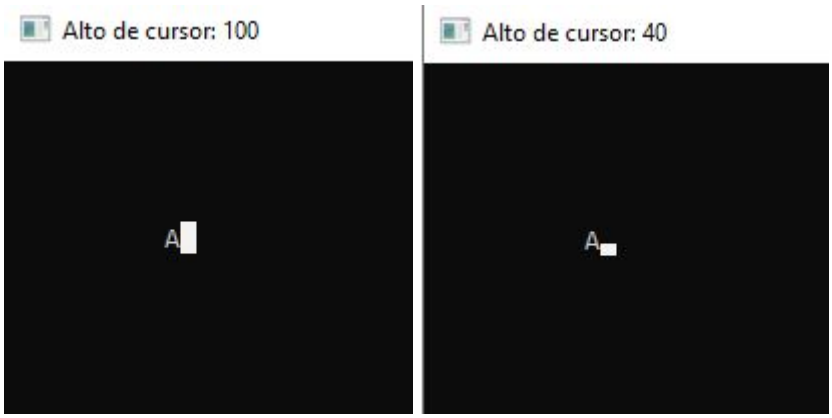
h es el handle a la consola

curInfo es una variable tipo CONSOLE_CURSOR_INFO

Ej.

```
CONSOLE_CURSOR_INFO cinfo;   // variable tipo CONSOLE_CURSOR_INFO
. . .
. . .
cinfo.dwSize = 50;    //altura del  cursor
cinfo.bVisible = TRUE;  //lo hace visible
SetConsoleCursorInfo(h,&cinfo);

. . .
. . .
// para ocultar el cursor
cinfo.dwSize = 1;
cinfo.bVisible = FALSE;
SetConsoleCursorInfo(h,&cinfo);
```



cursor con dwSize = 100 y cursor con dwSize = 40. Título de la ventana cambiado con setConsoleTitle

- `system( );`

Permite ejecutar un programa u orden externo.

ej.:

`system("cls");` borra la consola

`system("calc.exe");` ejecuta la calculadora de windows

Podés descargar ejemplos en:

http://progdispelec.weebly.com/uploads/1/0/2/3/102356212/color_forma_posicion.zip