

PDIH - UGR

Práctica 1 - Ángel Gómez Ferrer

Índice

Práctica 1 - Ángel Gómez Ferrer

Índice	1
Función gotoxy	2
Función setcursortype	2
Función setvideomode	3
Función getvideomode	3
Funciones textcolor y textbackgroundcolor	3
Función clrscr	4
Función cputchar	4
Función getche	4
Función para dibujar un recuadro en modo texto	5
Funcionamiento	6
Salidas del programa	7

Función gotoxy

Con esta función pondremos el cursor en la posición que queramos pasándole como parámetros x e y.

```
11 // EJ 1
12 void gotoxy(int x, int y) {
13     union REGS inregs, outregs;
14     inregs.h.ah = 2; // Indicamos que queremos cambiar la pos del cursor
15     inregs.h.dh = y; // Fila
16     inregs.h.dl = x; // Columna
17     inregs.h.bh = 0x00; // (En la ventana actual)
18     int86(0x10, &inregs, &outregs);
19 }
```

Función setcursortype

Nos permitirá fijar el aspecto del cursor como invisible, normal y grueso.

```
26 // EJ 2
27 void setcursortype(int cursor_type) {
28     union REGS inregs, outregs;
29     inregs.h.ah = 0x01;
30     switch(cursor_type){
31         case 0: //invisible
32             inregs.h.ch = 010;
33             inregs.h.cl = 000;
34             break;
35         case 1: //normal
36             inregs.h.ch = 010;
37             inregs.h.cl = 010;
38             break;
39         case 2: //grueso
40             inregs.h.ch = 000;
41             inregs.h.cl = 010;
42             break;
43     }
44     int86(0x10, &inregs, &outregs);
45 }
```

Función setvideomode

Nos permitirá cambiar el modo de video pasándole como parámetro el modo que queramos.

```
47 // EJ 3
48 void setvideomode(BYTE mode) {
49     union REGS inregs, outregs;
50     inregs.h.ah = 0;
51     inregs.h.al = mode;
52     int86(0x10, &inregs, &outregs);
53     return;
54 }
```

Función getvideomode

Nos permitirá obtener el modo actual de video.

```
56 // EJ 4
57 int getvideomode() {
58
59     int mode;
60     union REGS inregs, outregs;
61     inregs.h.ah = 15;
62     int86(0x10, &inregs, &outregs);
63     mode = outregs.h.al;
64     return mode;
65 }
```

Funciones textcolor y textbackgroundcolor

Utilizando unas variables globales “tcolor” y “bcolor” podremos cambiar con estas funciones el valor de las mismas para posteriormente usar esos valores en la función cputchar y cambiar el color del texto y el color de fondo de los caracteres que introduzcamos.

```
67 // EJ 5
68 void textcolor(unsigned char color) {
69     tcolor = color;
70 }
71 // EJ 6
72 void textbackground(unsigned char color) {
73     bcolor = color;
74 }
75
```

Función clrscr

Nos permitirá limpiar la pantalla.

Esta función si bien conseguí hacerla con interrupciones y la subfunción 0x06, me dio bastantes problemas y me parecía muy poco estable, tras investigar vi que 50 saltos de línea harían la misma función y evitaría así futuros problemas.

[illegible]

Función cputchar

Esta función nos permitirá mostrar un carácter por pantalla, como mencionamos anteriormente utilizaremos las variables “bcolor” y “tcolor” para asignarle un color de fondo y un color al carácter.

```

82 // EJ 8
83 void cputchar(char c) {
84     union REGS inregs, outregs;
85     inregs.h.ah = 0x09;
86     inregs.h.al = c;
87     inregs.h.bl = (bcolor << 4) | tcolor;
88     inregs.h.bh = 0x00;
89     inregs.x.cx = 1;
90
91     int86(0x10, &inregs, &outregs);
92 }

```

Función getche

Esta función nos permitirá obtener un carácter desde la entrada estándar y lo mostrará por pantalla.

```

95 // EJ 9
96 int getche() {
97     union REGS inregs, outregs;
98     int character;
99     inregs.h.ah = 1;
100    int86(0x21, &inregs, &outregs);
101    character = outregs.h.al;
102
103    return character;
104 }
105

```

Función para dibujar un recuadro en modo texto

Esta función recibe como parámetros las coordenadas superior izquierda e inferior derecha, también recibe el color de primer plano y el color de fondo.

Se compone de 4 bucles que se moverán por los ejes correspondientes dibujando cada uno de los lados del recuadro.

```
107 void dibujarcuadrado(int xsi, int ysi, int xid, int yid, unsigned char text, unsigned char bg) {
108
109     int i;
110
111     clrscr();
112     printf("\nPintar cuadrado");
113
114     textcolor(text);
115     textbackground(bg);
116     //Arriba
117     for (i = xsi; i <= xid; ++i) {
118         gotoxy(i, ysi);
119         cputchar(' ');
120     }
121     //Abajo
122     for (i = xsi; i <= xid; ++i) {
123         gotoxy(i, yid);
124         cputchar(' ');
125     }
126     //Izquierda
127     for (i = ysi; i <= yid; ++i) {
128         gotoxy(ysi, i);
129         cputchar(' ');
130     }
131     //Derecha
132     for (i = ysi; i <= yid; ++i) {
133         gotoxy(yid, i);
134         cputchar(' ');
135     }
136
137
138 }
```

Funcionamiento

Para comprobar el funcionamiento he creado un main que usa todas las funciones mencionadas anteriormente en el texto asociado a cada salida se puede ver los “ejercicios” (refiriéndome a las funciones en orden) a la que cada salida va asociado.

```
143 int main() {
144     int aux;
145     int video_mode;
146     clrscr();
147     printf("\nSe ha limpiado la pantalla.");
148     //Ej 1 Gotoxy
149     printf("\nEjercicio 1: Pulsa una tecla (GOTOXY 20,20): ");
150     gotoxy(20,20);
151     pause();
152
153     printf("\n");
154     // Ej 2 Cursor type
155     printf("\nEjercicio 2: Pulsa una tecla (Cursor invisible): ");
156     setcursortype(0);
157     pause();
158     printf("\nEjercicio 2: Pulsa una tecla (Cursor grueso): ");
159     setcursortype(2);
160     pause();
161     printf("\nEjercicio 2: Pulsa una tecla (Cursor normal): ");
162     setcursortype(1);
163     pause();
164     printf("\n");
165     // EJ 3 y 4 set video mode y get video mode
166     printf("\nEjercicio 3 y 4: Pulsa una tecla (setVideo): ");
167     setvideomode(MODOVIDEO);
168     video_mode = getvideomode();
169     printf("\nEjercicio 3 y 4: Pulsa una tecla (MODO GRAFICO 4): %i", video_mode);
170     pause();
171     printf("\nEjercicio 3 y 4: Pulsa una tecla (MODO TEXTO 3): ");
172     setvideomode(MODOTEXTO);
173     video_mode = getvideomode();
174     printf("\nEjercicio 3 y 4: (MODO TEXTO 3) video mode: %i", video_mode);
175     pause();
176     printf("\n");
177     //Ejs: 5 6 8 y 9
178     printf("\nEjercicios 5, 6, 8 y 9: Introduce una tecla: (GETCHE)");
179     aux = getch();
180     pause();
181     printf("\nEjercicios 5, 6, 8 y 9: Pulsa una tecla: ( TEXTCOLOR 1 (azul oscuro) y background 3 (cian) )\n");
182     pause();
183     textcolor(1);
184     textbackground(3);
185     cputchar((char)aux);
186     pause();
187     printf("\n");
188     //7 CLEAR
189     printf("\nTEXT0");
190     printf("\nEjercicios 7: Pulsa una tecla: (Limpiar Pantalla)\n");
191     pause();
192
193     pause();
194     dibujarcuadrado(1,1,10,10,1,3);
195
196
197
198
199     return 0;
200 }
201
```

Activ
Ve a Co

Salidas del programa

```
—  
  
Se ha limpiado la pantalla.  
Ejercicio 1: Pulsa una tecla (GOTOXY 20,20):
```

```
Ejercicio 2: Pulsa una tecla (Cursor invisible):
```

```
Ejercicio 2: Pulsa una tecla (Cursor grueso): █
```

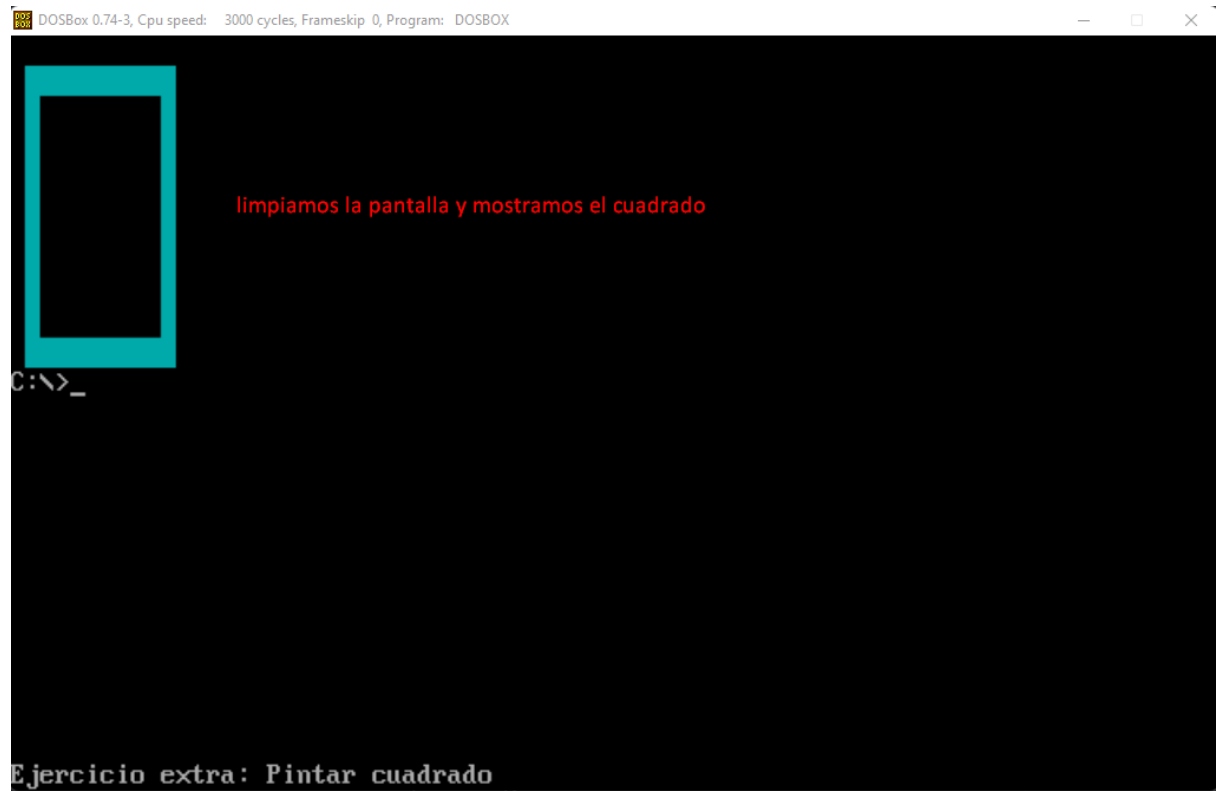
```
Ejercicio 2: Pulsa una tecla (Cursor normal): _
```

```
Ejercicio 3 y 4: Pulsa una tecla (MODO G  
RAFICO 4): 4 getvideomode()
```

```
Ejercicio 3 y 4: (MODO TEXTO 3) video mode: 3_
```

```
Ejercicios 5, 6, 8 y 9: Introduce una tecla: (GETCHE)D getche()  
Ejercicios 5, 6, 8 y 9: Pulsa una tecla: ( TEXTCOLOR 1 (azul oscuro) y backgroun  
d 3 (cyan) )  
D cputchar() con textcolor(1) y textbackground(2)
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1  
  
Ejercicio 3 y 4: (MODO TEXTO 3) video mode: 3  
  
Ejercicios 5, 6, 8 y 9: Introduce una tecla: (GETCHE)D  
Ejercicios 5, 6, 8 y 9: Pulsa una tecla: ( TEXTCOLOR 1 (azul oscuro) y backgroun  
d 3 (cyan) )  
D  
  
TEXTO  
Ejercicios 7: Pulsa una tecla: (Limpiar Pantalla)  
  
—  
  
antes de limpiar pantalla
```



Esas han sido las salidas que nos devuelve y la demostración de funcionamiento de cada una de ellas para el programa.