

PDIH - PRÁCTICA 1

Jorge Sánchez - Carlos Guasp



UNIVERSIDAD DE GRANADA

ÍNDICE

| | |
|---|----------|
| Código .c de las funciones | 2 |
| Ejecución de funciones a través del main() | 7 |

Código .c de las funciones

gotoxy(int x, int y): Mueve el cursor a la posición (x, y) en pantalla usando la interrupción 0x10 de DOS.

```
// gotoxy(): coloca el cursor en una posición determinada
void gotoxy(int x, int y) {
    union REGS inregs, outregs;

    inregs.h.dh = y;
    inregs.h.dl = x;
    inregs.h.bh = 0;
    inregs.h.ah = 0x02;

    int86(0x10, &inregs, &outregs);
}
```

setcursortype(int type): Cambia la apariencia del cursor ajustando los registros **ch** y **cl** según el tipo (invisible, normal o grueso) mediante la interrupción 0x10.

```
// setcursortype(): fijar el aspecto del cursor
void setcursortype(int type) {
    union REGS inregs, outregs;

    inregs.h.ah = 0x01;

    switch (type) {
        case 0: // INVISIBLE
            inregs.h.ch = 0x20;
            inregs.h.cl = 0x00;
            break;
        case 1: // NORMAL
            inregs.h.ch = 0x0C;
            inregs.h.cl = 0x0D;
            break;
        case 2: // THICK
            inregs.h.ch = 0x00;
            inregs.h.cl = 0x07;
            break;
        default:
            return;
    }

    int86(0x10, &inregs, &outregs);
}
```

setvideomode(int mode): Establece el modo de video enviando el valor de **mode** en **al** a la interrupción 0x10.

```
// setvideomode(): fija el modo de video deseado
void setvideomode(int mode) {
    union REGS inregs, outregs;

    inregs.h.ah = 0x00;
    inregs.h.al = mode;

    int86(0x10, &inregs, &outregs);
}
```

getvideomode(): Obtiene el modo de video actual llamando a la interrupción 0x10 con **ah = 0x0F** y devuelve el valor de **al**.

```
// getvideomode(): obtiene el modo de video actual
int getvideomode() {
    union REGS inregs, outregs;

    inregs.h.ah = 0x0F;

    int86(0x10, &inregs, &outregs);

    return outregs.h.al;
}
```

textcolor(int color): Cambia el color del texto guardándolo en **currentTextColor**, siempre que esté en el rango 0-15.

```
unsigned char currentTextColor = 7; // Color del primer plano por defecto (blanco)
// textcolor(): modifica el color de primer plano
void textcolor(int color) {
    if (color >= 0 && color <= 15) {
        currentTextColor = color;
    }
}
```

textbackground(int color): Cambia el color de fondo guardándolo en **currentBackgroundColor**, si está entre 0-15.

```
unsigned char currentBackgroundColor = 0; // Color de fondo por defecto (negro)
// textbackground(): modifica el color de fondo
void textbackground(int color) {
    if (color >= 0 && color <= 15) {
        currentBackgroundColor = color;
    }
}
```

clrscr(): Limpia la pantalla desplazando todo su contenido usando la interrupción 0x10 y recoloca el cursor en (0,0).

```
// clrscr(): borra toda la pantalla
void clrscr() {
    union REGS inregs, outregs;

    // Desplazamiento de pantalla, se desea limpiar la pantalla.
    inregs.h.ah = 0x06;      // Función para desplazar la ventana de video
    inregs.h.al = 0x00;      // Número de líneas a desplazar (0 para limpiar)
    inregs.h.bh = currentBackgroundColor; // Color de fondo
    inregs.h.ch = 0x00;      // Fila superior
    inregs.h.cl = 0x00;      // Columna izquierda
    inregs.h.dh = 24;        // Fila inferior (24 es para 25 filas)
    inregs.h.dl = 79;        // Columna derecha (79 es para 80 columnas)

    // Llamada a la interrupción
    int86(0x10, &inregs, &outregs);

    // Mueve el cursor a la posición inicial (0,0)
    gotoxy(0, 0);
}
```

cputchar(char c): Escribe un carácter en pantalla con los colores actuales llamando a la interrupción 0x10.

```
// cputchar(): escribe un carácter en pantalla con el color indicado actualmente
void cputchar(char c) {
    union REGS inregs, outregs;
    inregs.h.ah = 0x09;
    inregs.h.al = c;
    inregs.h.bl = (currentTextColor << 4) | currentBackgroundColor;
    inregs.h.bh = 0x00;
    inregs.x.cx = 1;

    int86(0x10, &inregs, &outregs);
}
```

getche(): Captura un carácter desde el teclado y lo muestra en pantalla mediante la interrupción 0x21.

```
// getche(): obtiene un carácter de teclado y lo muestra en pantalla
int getche() {
    union REGS inregs, outregs;
    int character;
    inregs.h.ah = 0x01;

    int86(0x21, &inregs, &outregs);
    character = outregs.h.al;

    return character;
}
```

pixel(int x, int y, unsigned char color): Dibuja un píxel en (x, y) con el color especificado en modo gráfico usando la interrupción 0x10.

```
// pixel(): dibujar un píxel en modo gráfico
void pixel(int x, int y, unsigned char color) {
    union REGS inregs, outregs;

    inregs.h.ah = 0x0C;
    inregs.h.al = color;
    inregs.x.cx = x;
    inregs.x.dx = y;
    inregs.h.bh = 0;

    int86(0x10, &inregs, &outregs);
}
```

stop(): Detiene la ejecución esperando una tecla con la interrupción 0x16.

```
// stop(): función para pausas en el main
void stop() {
    union REGS inregs, outregs;
    inregs.h.ah = 0x00;
    int86(0x16, &inregs, &outregs);
}
```

drawBox(int x1, int y1, int x2, int y2): Dibuja un rectángulo en pantalla usando gotoxy() y cputchar().

```
// drawBox(): dibuja un cuadrado en la pantalla
void drawBox(int x1, int y1, int x2, int y2) {
    int i;

    for (i = x1; i <= x2; ++i) {
        gotoxy(i, y1);
        cputchar(' ');
        gotoxy(i, y2);
        cputchar(' ');
    }

    for (i = y1; i <= y2; ++i) {
        gotoxy(x1, i);
        cputchar(' ');
        gotoxy(x2, i);
        cputchar(' ');
    }
}
```

drawLine(int x1, int y1, int x2, int y2): Dibuja una línea en la pantalla mediante el algoritmo de interpolación de puntos, usando `gotoxy()` y `cputchar()`.

```
// drawLine(): dibuja una línea en la pantalla
void drawLine(int x1, int y1, int x2, int y2) {
    int dx = x2 - x1;
    int dy = y2 - y1;
    int steps = (abs(dx) > abs(dy)) ? abs(dx) : abs(dy);
    int x, y, i;
    float xIncrement = (float)dx / (float)steps;
    float yIncrement = (float)dy / (float)steps;

    x = x1;
    y = y1;
    for (i = 0; i <= steps; i++) {
        gotoxy(x, y);
        cputchar('x');
        x += (int)xIncrement;
        y += (int)yIncrement;
    }
}
```

drawAsciiArt(): Muestra un menú para elegir entre tres dibujos ASCII (perro, gato o pato) y los imprime en pantalla con `printf()`.

```
//drawAsciiArt(): dibuja un arte ASCII en la pantalla
void drawAsciiArt() {
    int option;
    printf("1. Perro\n");
    printf("2. Gato\n");
    printf("3. Pato\n");
    printf("Ingrese un numero del 1 al 3: ");
    scanf("%d", &option);
    getchar();

    switch(option) {
        case 1:
            printf("Rombo: \n");
            printf("  /  \ \n");
            printf(" / _ \ \n");
            printf("| /  \ | \n");
            printf("|  \ /  | \n");
            printf(" \ \  / \n");
            printf("  \_ / \n");
            break;

        case 2:
            printf("Gato: \n");
            printf("/\\_\\ \n");
            printf("( o.o ) \n");
            printf("> ^ < \n");
            break;

        case 3:
            printf("Pato: \n");
            printf(" __ \n");
            printf("<(o) \n");
            printf(" || \n");
            break;

        default:
            printf("Opción no válida. Por favor, ingrese un número del 1 al 3.\n");
            break;
    }
}
```

Finalmente, el **main()** muestra un menú que permite probar cada función, solicitando al usuario una opción y ejecutando la función correspondiente.

```
int main() {
    int option;
    int aux;
    printf("Ejercicios de la practica 1 Jorge y Carlos: \n");
    printf("1. gotoxy(10,10)\n");
    printf("2. setcursortype()\n");
    printf("3. setvideomode() y getvideomode()\n");
    printf("4. textcolor(), textbackground(), cputchar() y\n");
    getch()\n");
    printf("5. pixel()\n");
    printf("6. drawBox()\n");
    printf("7. drawLine()\n");
    printf("8. drawAssciiArt()\n");
    printf("Ingrese un numero del 1 al 8: ");
    scanf("%d", &option);
    getchar(); // Limpiar el buffer de entrada

    switch(option) {
        case 1:
            printf("Ejercicio 1: gotoxy(10,10): \n");
            gotoxy(10, 10);
            stop();
            break;

        case 2:
            printf("Ejercicio 2: setcursortype(): \n");
            printf("Cursor invisible: \n");
            setcursortype(0);
            stop();
            printf("Cursor grueso: \n");
            setcursortype(2);
            stop();
            printf("Cursor normal: \n");
            setcursortype(1);
            stop();
            break;

        case 3:
            printf("Ejercicios 3 y 4: setvideomode() y getvideomode():\n");
```



```
        printf("Modo de video actual: %d\n", getvideomode());
        setvideomode(0x04);
        printf("Modo de video actual: %d\n", getvideomode());
        stop();
        break;

    case 4:
        printf("Ejercicios 5, 6, 7, 8 y 9: textcolor(),
textbackground(), clrscr(), cputchar() y getche(): \n");
        printf("Introduce el caracter:");

        aux = getche();
        printf("\n");

        stop();
        printf("Color de texto: 1 (azul oscuro) y color de fondo:
10 (verde claro)\n");
        textcolor(10);
        textbackground(1);

        printf("Caracter introducido, con cambios: ");
        cputchar((char)aux);
        stop();

        clrscr();
        stop();
        break;

    case 5:
        printf("Ejercicio 10: pixel(): \n");
        setvideomode(0x04);
        printf("Dibujando pixeles...\n");
        pixel(10, 10, 1);
        stop();
        break;

    case 6:
        clrscr();
        printf("Ejercicio 11: drawBox(): \n");
        textcolor(10);
        textbackground(1);
        drawBox(5, 5, 15, 15);
        stop();
```

```
        stop();
        stop();
        break;

    case 7:
        clrscr();
        printf("Ejercicio 12: drawLine(): \n");
        textcolor(10);
        textbackground(1);
        setvideomode(0x04);
        printf("Dibujando lineas...\n");
        drawLine(5, 5, 15, 15);
        stop();
        stop();
        stop();
        stop();
        stop();
        break;

    case 8:
        clrscr();
        printf("Ejercicio 13: drawAssciiArt(): \n");
        textcolor(10);
        textbackground(1);
        drawAssciiArt();
        stop();
        stop();
        stop();
        break;

    default:
        printf("Opción no válida. Por favor, ingrese un número del
1 al 5.\n");
        break;
}

return 0;
}
```

Ejecución de funciones a través del `main()`

Ej 1

```
C:\TURBOC3\BIN>TC C:\TURBOC3\Projects\P1.C  
Ejercicio 1: gotoxy (10,10):
```

-

Ej 2

```
C:\TURBOC3\BIN>TC C:\TURBOC3\Projects\P1.c  
Ejercicio 2: setcursortype():  
Cursor invisible:
```

```
C:\TURBOC3\BIN>TC C:\TURBOC3\Projects\P1.c  
Ejercicio 2: setcursortype():  
Cursor invisible:  
Cursor grueso: █
```

```
Cursor invisible:  
Cursor grueso:  
Cursor normal: _
```

Ej 3 y 4

```
Modo de video actual: 4
```

Ej 5,6,8 y 9:

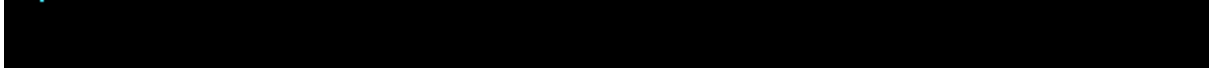
```
C:\TURBOC3\BIN>TC C:\TURBOC3\Projects\P1.c  
Ejercicio 5, 6, 8 y 9: textcolor(), textbackground(), cputchar() y getche():  
Introduce el caracter:a  
Color de texto: 1 (azul oscuro) y color de fondo: 10 (verde claro)  
Caracter introducido, con cambios: a
```

Ej 7, justo después de lo anterior:



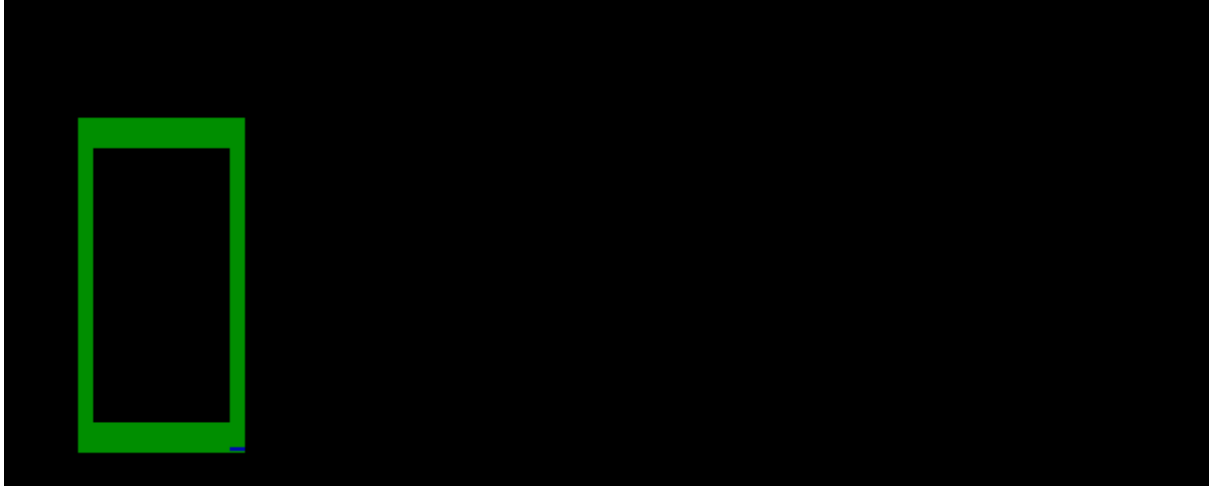
Ej 10:

Dibujando pixeles...



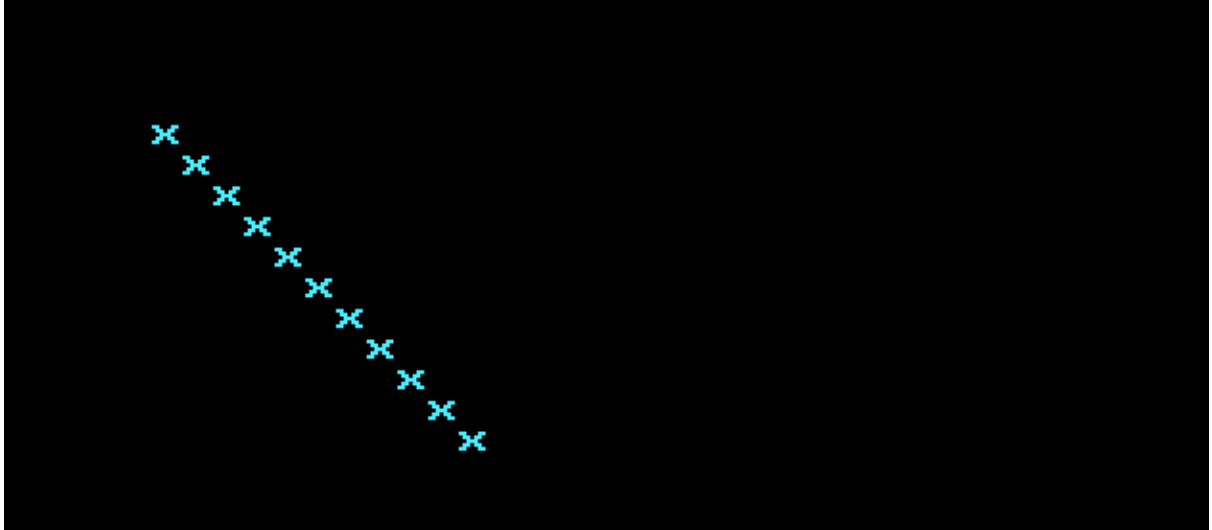
Ej 11:

Ejercicio 11: drawBox():



Ej 12:

Dibujando lineas...



Ej 13:

```
Ejercicio 13: drawAsciiArt():  
1. Perro  
2. Gato  
3. Pato  
Ingrese un numero del 1 al 3: 2  
Gato:  
  ^_^  
( o.o )  
> ^ <
```