

# **Práctica 1. Entrada/Salida utilizando interrupciones con lenguaje C**



**UNIVERSIDAD  
DE GRANADA**

Marta Díaz Artigot

## Variables y funciones auxiliares

Variables utilizadas en las funciones.

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

#define BYTE unsigned char

BYTE MODOTEXTO = 3;
BYTE MODOGRAFICO = 4;

unsigned char color_fondo = 0;
unsigned char color_texto = 1;
```

Función Pausa()

```
// Función Pausar guión
void pausa(void){
    union REGS inregs, outregs;
    inregs.h.ah = 0x00;
    int86(0x16, &inregs, &outregs);
}
```

## Requisitos mínimos

Código de cada función e imágenes de la prueba de funcionamiento.

### 1. Función gotoxy()

```
// Función para colocar el cursor en una posición determinada
void gotoxy(int x, int y){
    union REGS inregs, outregs;
    inregs.h.ah = 0x02;
    inregs.h.dh = x; //Fila
    inregs.h.dl = y; //Columna
    inregs.h.bh = 0x00;
    int86(0x10, &inregs, &outregs);
}
```

\*\*\*\*\*

Colocar el cursor en (10, 10).

## 2. Función setcursortype()

```
// Función para fijar el aspecto del cursor, debe admitir tres valores: INVISIBLE, NORMAL y GRUESO.
void setcursortype(int tipo_cursor){
    union REGS inregs, outregs;
    inregs.h.ah = 0x01;

    switch (tipo_cursor){
        case 0: //INVISIBLE
            inregs.h.ch = 010;
            inregs.h.cl = 000;
            break;
        case 1: //NORMAL
            inregs.h.ch = 010;
            inregs.h.cl = 010;
            break;
        case 2: //GRUESO
            inregs.h.ch = 000;
            inregs.h.cl = 010;
            break;
    }

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

DOS BOX DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1

Fijamos el cursor en modo GRUESO.

```
Fijamos el cursor en modo GRUESO.  
Fijamos el cursor en modo NORMAL.  
—
```

```
Fijamos el cursor en modo GRUESO.  
Fijamos el cursor en modo NORMAL.  
Fijamos el cursor en modo INVISIBLE.
```

### 3. Función setvideomode()

```
// Función para fijar el modo de video deseado  
void setvideomode(BYTE modo){  
    union REGS inregs, outregs;  
    inregs.h.al = modo;  
    inregs.h.ah = 0x00;  
    int86(0x10, &inregs, &outregs);  
}
```

### 4. Función getvideomode()

```
// Función para obtener el modo de video actual  
BYTE getvideomode(void){  
    union REGS inregs, outregs;  
    int modo;  
    inregs.h.ah = 15;  
    int86(0x10, &inregs, &outregs);  
    modo = outregs.h.al;  
    return modo;  
}
```

```
Obtenemos el modo de video actual:  
4
```

```
Obtenemos el modo de video actual:  
3
```

## 5. Función textcolor()

```
// Función para modificar el color de primer plano con que se mostrarán los caracteres
void textcolor(unsigned char color){
    color_texto = color;
}
```

## 6. Función textbackground()

```
// Función para modificar el color de fondo con que se mostrarán los caracteres
void textbackground(unsigned char color){
    color_fondo = color;
}
```

Imprimimos con los nuevos colores de texto y de fondo:

15

## 7. Función clrscr()

[illegible]

```

Available memory 4166296
C:\P1-EJE~1>P1.EXE
Comprobacion de funciones:

```

Se ha limpiado la pantalla.

## 8. Función cputchar()

```
// Función para escribir un carácter en pantalla con el color indicado actualmente
void cputchar(char caracter){
    union REGS inregs, outregs;
    inregs.h.ah = 0x09;
    inregs.h.al = caracter;
    inregs.h.bl = color_fondo << 4 | color_texto;
    inregs.h.bh = 0x00;
    inregs.x.cx = 1;
    int86(0x10, &inregs, &outregs);
}
```

## 9. Función getche()

```
// Función para obtener un carácter de teclado y lo muestra en pantalla
int getche(void){
    union REGS inregs, outregs;
    int caracter;
    inregs.h.ah = 1;
    int86(0x21, &inregs, &outregs);
    caracter = outregs.h.al;

    return caracter;
}
```

Pulsa una tecla:

La tecla que has pulsado es:



## Main

Función Main para probar el código.

```
int main() {
    BYTE modo;
    int car;
    printf("Comprobacion de funciones:\n");
    pausa();
    clrscr();
    printf("Se ha limpiado la pantalla.\n");
    pausa();
    clrscr();

    printf("Colocar el cursor en (10, 10).\n");
    gotoxy(10,10);
    printf("*****\n");
    pausa();
    clrscr();

    gotoxy(0,0);
    printf("Fijamos el cursor en modo GRUESO.\n");
    setcursortype(2);
    pausa();
    printf("Fijamos el cursor en modo NORMAL.\n");
    setcursortype(1);
    pausa();
    printf("Fijamos el cursor en modo INVISIBLE.\n");
    setcursortype(0);
    pausa();

    //Volvemos al normal
    setcursortype(1);

    textcolor(9);
    textbackground(5);
    printf("Imprimimos con los nuevos colores de texto y de fondo:\n");
    putchar('c');
    pausa();
}
```

```
printf("Establecemos el modo texto de video:\n");
setvideomode(4);
pausa();

printf("Obtenemos el modo de video actual:\n");
modo = getvideomode();
printf("%d\n",modo);
pausa();

printf("Establecemos el modo gráfico de video:\n");
setvideomode(3);
pausa();

printf("Obtenemos el modo de video actual:\n");
modo = getvideomode();
printf("%d\n",modo);
pausa();

printf("Pulsa una tecla:\n");
car = getche();
printf("\nLa tecla que has pulsado es:\n");
cputchar(car);
pausa();
```



# Requisitos Ampliados

## 1. Función dibujar\_recuadro()

```
// Función para a dibujar un recuadro en la pantalla en modo texto. Recibirá como parámetros las coordenadas
// el color de primer plano y el color de fondo.
void dibujar_recuadro(int x1, int y1, int x2, int y2, int color_frente, int color_fondo) {
    int i, j;

    // Dibujar borde superior
    for (i = x1; i <= x2; i++) {
        gotoxy(i, y1);
        textcolor(color_frente);
        putchar(219); // Caracter de borde horizontal
    }

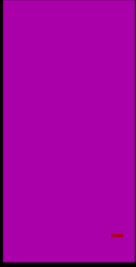
    // Dibujar borde inferior
    for (i = x1; i <= x2; i++) {
        gotoxy(i, y2);
        textcolor(color_frente);
        putchar(219); // Caracter de borde horizontal
    }

    // Dibujar borde izquierdo
    for (j = y1 + 1; j <= y2 - 1; j++) {
        gotoxy(x1, j);
        textcolor(color_frente);
        putchar(179); // Caracter de borde vertical
    }

    // Dibujar borde derecho
    for (j = y1 + 1; j <= y2 - 1; j++) {
        gotoxy(x2, j);
        textcolor(color_frente);
        putchar(179); // Caracter de borde vertical
    }

    // Rellenar interior del recuadro
    for (i = x1 + 1; i <= x2 - 1; i++) {
        for (j = y1 + 1; j <= y2 - 1; j++) {
            gotoxy(i, j);
            textcolor(color_fondo);
            putchar(' ');
        }
    }
}
```

Dibujamos un recuadro:



## Main

```
clrscr();  
gotoxy(0,0);  
printf("\nDibujamos un recuadro:\n");  
dibujar_recuadro(10, 5, 20, 15, 5, 4);  
pausa();
```