PRACTICA 1

Entrada/Salida utilizando interrupciones con lenguaje C

Pablo Fernández Tello

En esta práctica hemos utilizado las rutinas de servicio de interrupción de la BIOS para teclado y vídeo desde el sistema operativo MS-DOS para programar las funciones que se nos piden.

Todas ellas están implementadas en el mismo archivo .c y al final de este he hecho un pequeño main que prueba cada una de estas funciones. Aun así, mostraré algunas capturas de el funcionamiento de estas.

Funciones auxiliares

Primero voy a describir las funciones auxiliares que utilizan muchas de las funciones que he implementado:

-mi_pausa(): esta función pausa la ejecución del programa hasta que se presiona una tecla. Utiliza la interrupción 0x21 del sistema DOS con el servicio 0x08 para esperar a que se presione una tecla.

mi_getchar(): esta función lee un carácter desde la entrada estándar y lo devuelve como un entero. Llama a la función int86() con la interrupción 0x21 para invocar el servicio 1 del sistema DOS y leer el carácter.

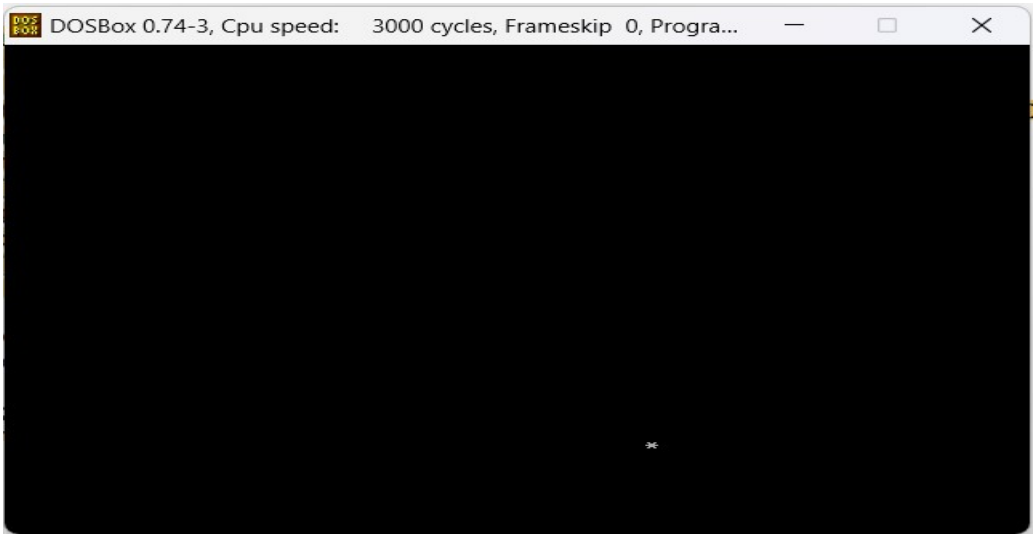
mi_putchar(char c): esta función llama a la función int86() con la interrupción 0x21 para invocar el servicio del sistema DOS y escribir el carácter en la salida estándar.

escribir_char_con_color(char caracter): esta función escribe un carácter en la pantalla de texto con un color dado. Utiliza la interrupción 0x10 de la BIOS con el servicio 0x09 para escribir el carácter con el color especificado.

Funciones principales

- gotoxy(): esta función mueve el cursor a una posición especificada en la pantalla. Utiliza la subfunción 0x02 de la interrupción 0x10 (INT 10h) de la BIOS de la tarjeta gráfica para controlar la posición del cursor.

He movido el cursos a la posición 50,20



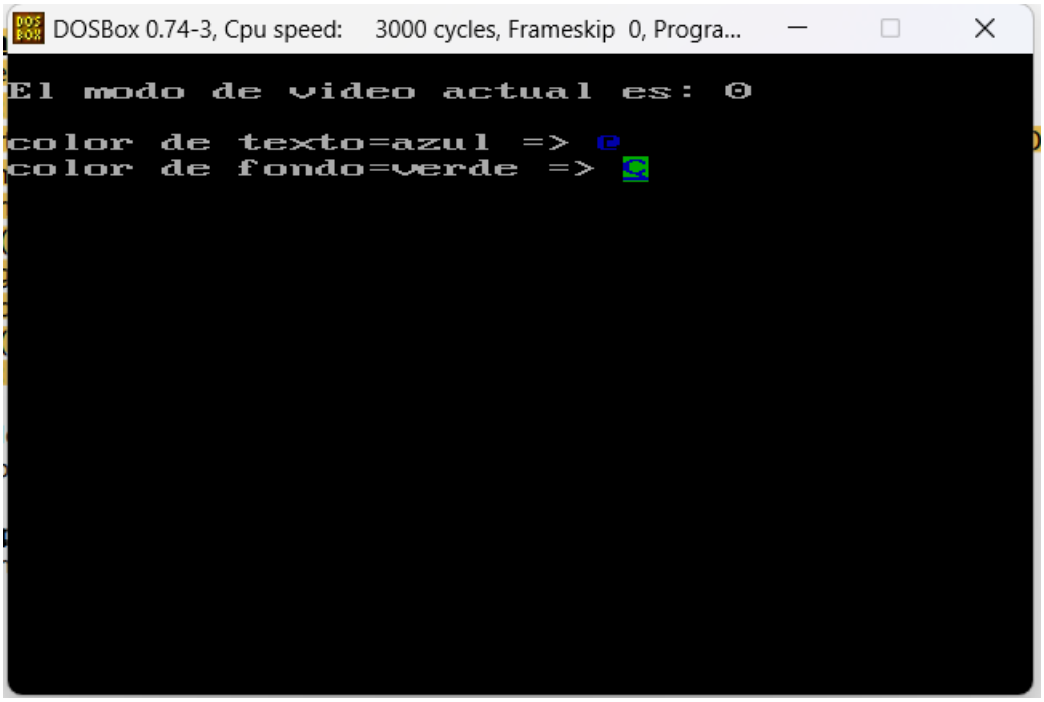
- **setcursortype():** esta función establece el tipo de cursor en la pantalla. Utiliza la subfunción 0x01 de la interrupción 0x10 (INT 10h) de la BIOS de la tarjeta gráfica para cambiar el tipo de cursor. En este caso, nos piden que podamos cambiar el cursor a invisible, normal y grueso.

- **setvideomode():** esta función establece el modo de video deseado en la pantalla. Utiliza la subfunción 0x00 de la interrupción 0x10 (INT 10h) de la BIOS de la tarjeta gráfica para cambiar el modo de video.

- **getvideomode():** esta función obtiene el modo de video actual de la pantalla. Utiliza la subfunción 0x0F de la interrupción 0x10 de la BIOS de la tarjeta gráfica para leer el modo de video actual.

- **textcolor():** esta función modifica el color de primer plano con que se mostrarán los caracteres en la pantalla. Para ello, actualiza la variable global `ctexto` con el color deseado.

- **textbackground():** esta función modifica el color de fondo con que se mostrarán los caracteres. Para ello, actualiza la variable global `cfondo` con el color deseado.



- **clrscr():** Esta función simplemente imprime por pantalla un salto de linea 25 veces, que corresponde al tamaño de esta y borra toda la pantalla. A continuación, coloca el cursos en la posición 0,0.

- cputchar(): esta función escribe un carácter en pantalla con el color indicado actualmente. Para ello llama a la funcion textcolor que actualiza el color de texto y a la función auxiliar escribir_char_con_color que escribe un char con el color de texto y de fondo que indiquen las variables globales `ctexto` y `cfondo`.

Esta función auxiliar se describe más adelante.

- getche(): obtiene un carácter de teclado y lo muestra en pantalla. Para ello se apoya en la función auxiliar mi_getchar que obtiene el carácter del teclado y en la función mi_putchar que muestra el carácter por pantalla.

Ejercicio opcional

Por último he implementado una función llamada dibujaRecuadro para el ejercicio opcional en el que nos piden dibujar un recuadro en modo texto.

Para ello primero he llamado a las funciones textcolor y textbackground para definir el color de texto y el de fondo deseado y a continuación he procedido de la siguiente manera:

Primero he representado las aristas superior e inferior recorriendo las posiciones correspondientes marcadas por un bucle for con la función gotoxy y he ido imprimiendo un carácter por cada una de esas posiciones con la función cputchar.

Por último he hecho lo mismo con las aristas laterales.