

PRÁCTICA 1. ENTRADA/SALIDA UTILIZANDO INTERRUPCIONES CON LENGUAJE C

Periféricos y Dispositivos de Interfaz Humana

Grado en Ingeniería informática
Curso 2021-2022



UNIVERSIDAD DE GRANADA

Santiago Gil Legaza
Victoria Dueñas Salcedo

1. REQUISITOS MÍNIMOS

1.1. Colocar el cursor en una posición determinada

Esta primera función recibe como parámetros dos *int*, uno de ellos para la coordenada X y el otro para la coordenada Y. La coordenada (X,Y) será la posición determinada a la que se llevará el cursor. La función no tiene salida.

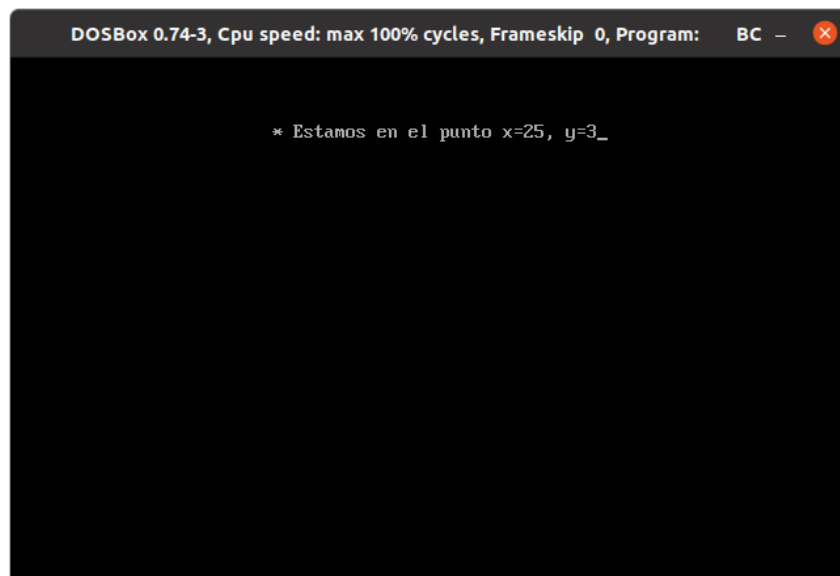
La interrupción que necesitamos para llevar a cabo esta operación es la interrupción de la BIOS número 10h (controla los servicios de pantalla del PC).

En el registro AH indicamos la función a la que deseamos acceder. En este caso, la función número 2.

En el registro BH añadimos el valor 0.

Por último, los registros DH y DL almacenan el número de la columna y el número de la fila, respectivamente.

Con las coordenadas (25,3), el resultado obtenido sería el siguiente:



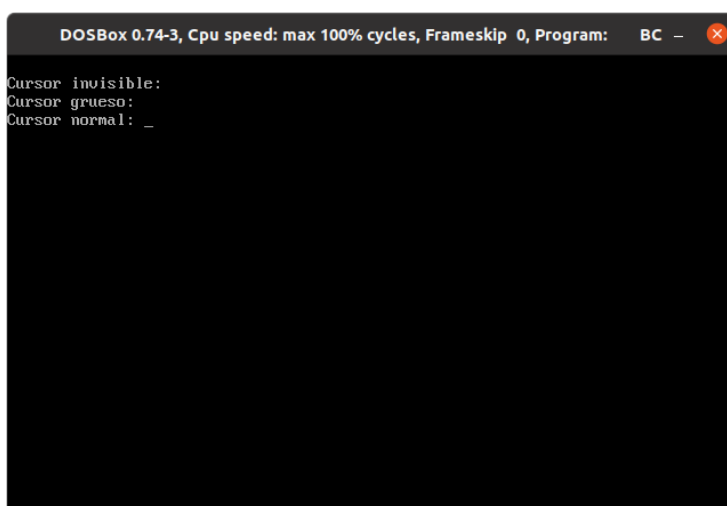
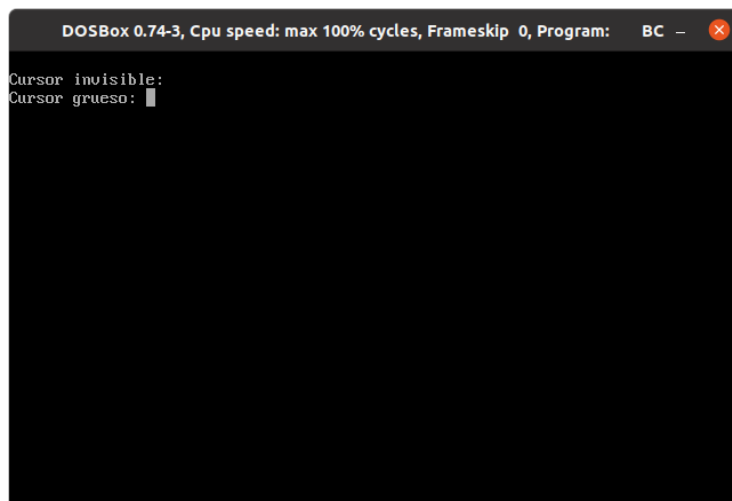
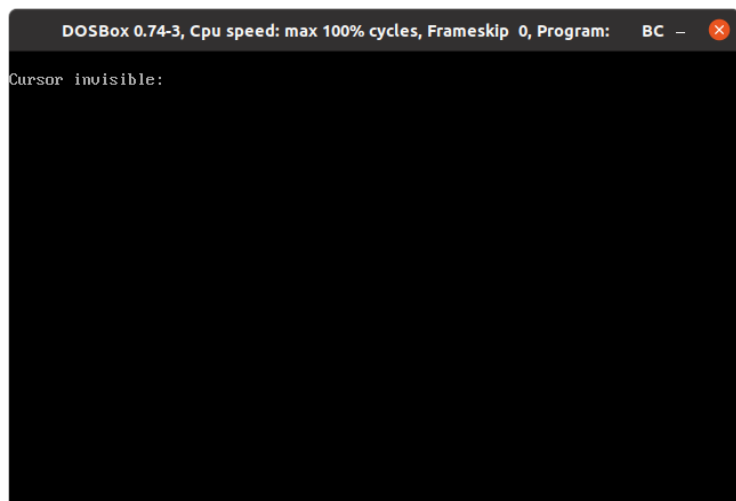
1.2. Fijar el aspecto del cursor

La segunda función muestra por pantalla tres diferentes tipos de cursor: invisible, normal y grueso. Como parámetro recibe el tipo de cursor, siendo 0 el cursor invisible, 1 el cursor normal y 2 el cursor grueso. Esta función no tiene salida.

La interrupción que necesitamos para llevar a cabo esta operación es la interrupción número 10h, al igual que en la función anterior. En el registro AH indicamos que queremos acceder a la función número 1.

En los registros CH y CL indicamos los tres bits menos significativos del número de la línea inicial y el número de la línea final, respectivamente. Para el cursor invisible, sería ch=010 y cl=000. Para el cursor normal, ch=010 y cl=010. Y para el cursor grueso, ch=000 y cl=010.

Vemos a continuación los diferentes tipos de cursor.



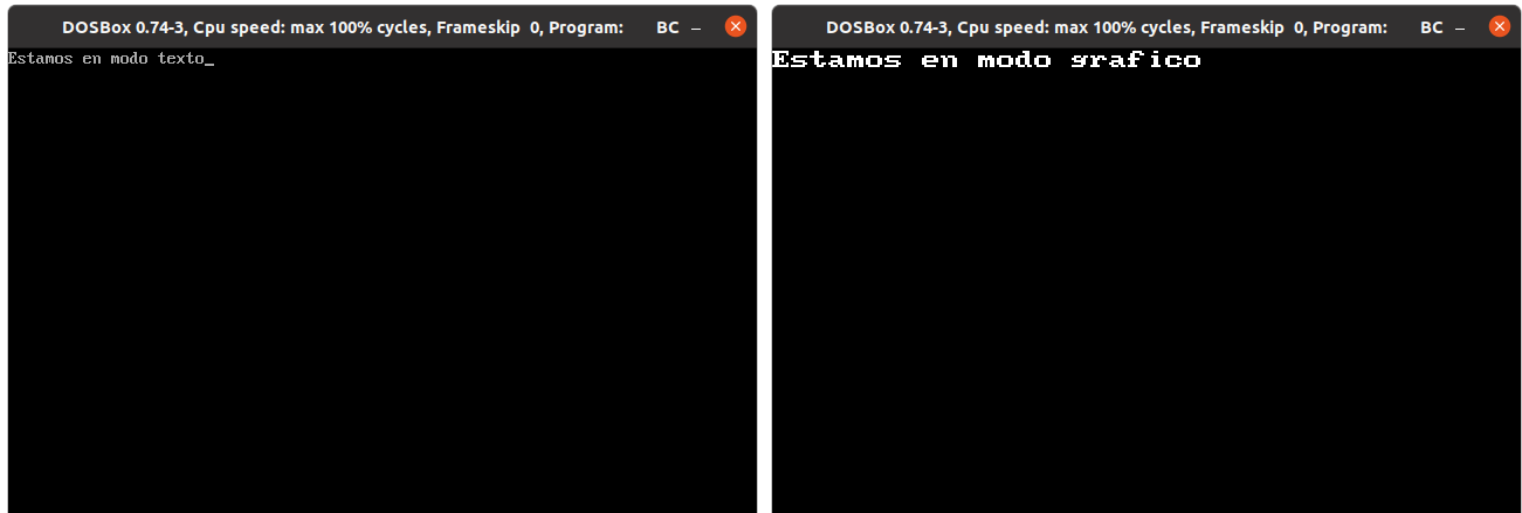
1.3. Fijar el modo de video deseado

Se encarga de cambiar el modo de video. Recibe como parámetro el *modo*, almacenado en el registro AL. El número de función, que se va a almacenar en el registro AH, es 0. Al igual que las dos anteriores funciones, no tiene salida y utiliza la interrupción 10h. Si queremos que esté en modo texto, el parámetro *modo* tiene que ser 0, 1, 2, 3 o 7. Por el contrario, para que esté en modo gráfico, tendría que ser 4, 5, 6, D, E, F, 12 y 13.

1.4. Obtener el modo de video actual

La función creada para obtener el modo de video no recibe parámetros ni tampoco tiene ninguna salida. En esta función volvemos a usar la interrupción 10h, pero esta vez necesitaremos utilizar la función Fh.

Vamos a ver un ejemplo en el que vemos el funcionamiento de la función de obtener el modo de video actual, pero también vemos cómo funciona la función que fija el modo de video deseado. Si estamos en modo texto, se muestra un mensaje que dice “Estamos en modo texto”. Por el contrario, si estamos en modo gráfico, se muestra por pantalla “Estamos en modo gráfico”.



1.5. Modificar el color de primer plano y de fondo con que se mostrarán los caracteres

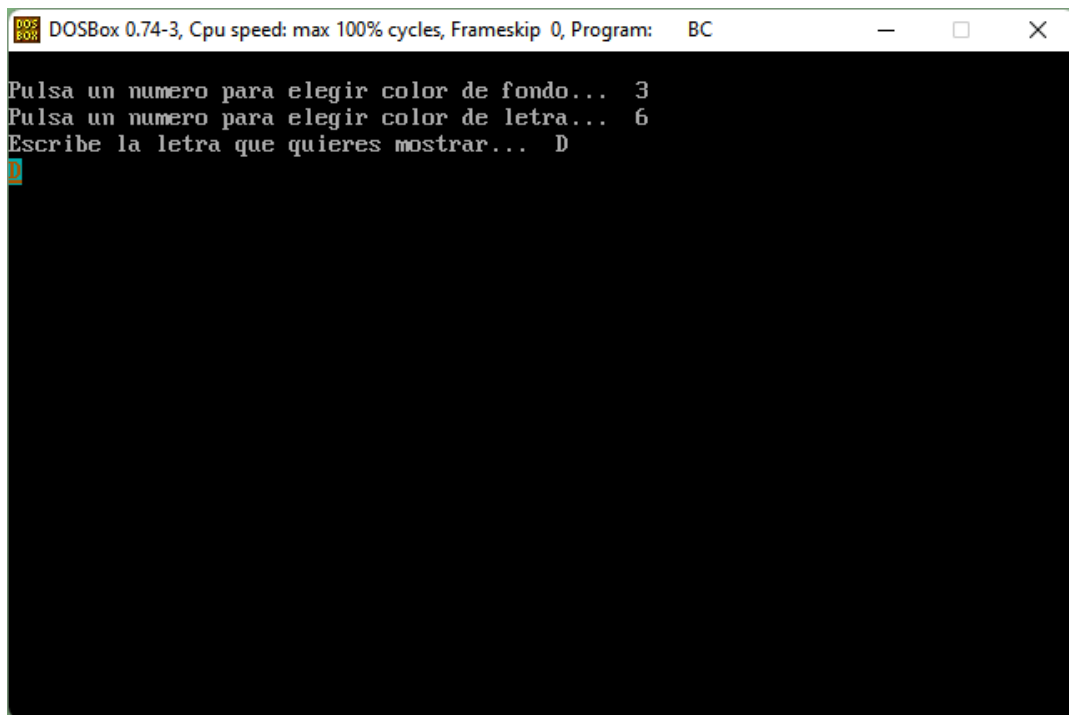
En este caso, tenemos en el main una parte en la que introducimos por teclado el número del color que queremos para el carácter. También se lee por teclado el número del color que queremos para el fondo y finalmente lee el carácter que queremos mostrar. Ambos colores se guardan en unas variables globales que después serán usadas para mostrar el carácter con la función 1.7. Para leer los valores de teclado usamos la función 1.8.

1.6. Borrar toda la pantalla

Para esta función no hacen falta parámetros de entrada. Solamente usa el registro AH, primero con el valor de función 15, posteriormente llama a la interrupción de la BIOS 10h. Ahora vuelve a usar el registro AH, pero esta vez con el valor de función 0 y llama de nuevo a la interrupción 10h.

1.7. Escribir un carácter en pantalla con el color indicado actualmente

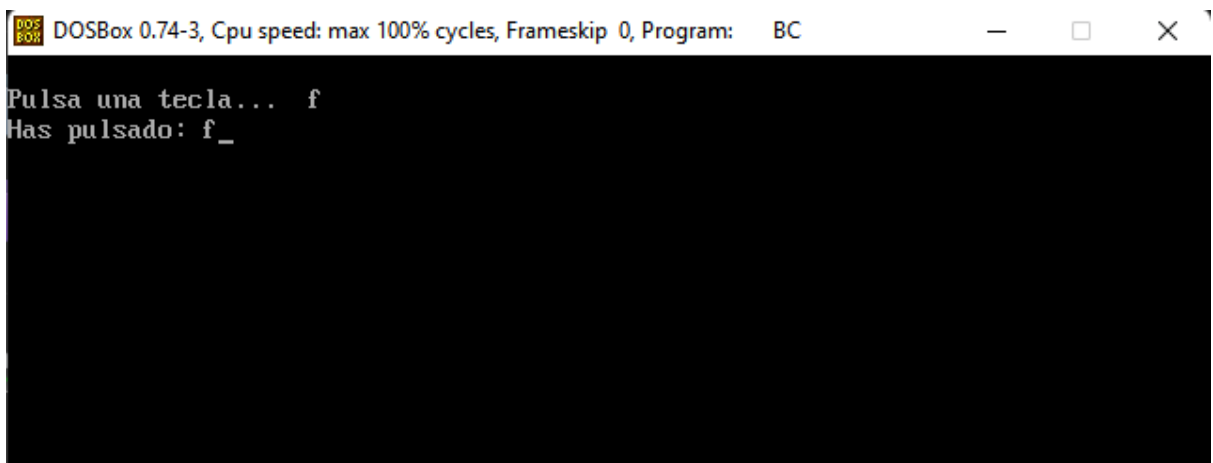
En esta función tenemos como entrada el carácter que queremos representar por pantalla. La función hace uso de la interrupción 10h y usa 5 registros diferentes para realizar su función. El registro AH con el valor 9, el registro AL en el que se introduce el código ASCII del carácter, el registro BL en el que se ingresa el color de fondo y del carácter, el registro BH con el valor 0 y el registro CX en el que se ingresa el número de repeticiones, y en este caso es 1. La peculiaridad del registro BL es que es de 1 byte, y con los primeros cuatro bits se fija el color de fondo y con los últimos 4 bits se fija el color del carácter.



```
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Pulsa un numero para elegir color de fondo... 3
Pulsa un numero para elegir color de letra... 6
Escribe la letra que quieres mostrar... D
D
```

1.8. Obtener un carácter de teclado y mostrarlo en pantalla

Para esta función tenemos el registro AH con el valor de función 1 y usa la interrupción DOS 21h para el teclado. Posteriormente devuelve el carácter leído en el registro AL.



```
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Pulsa una tecla... f
Has pulsado: f_
```

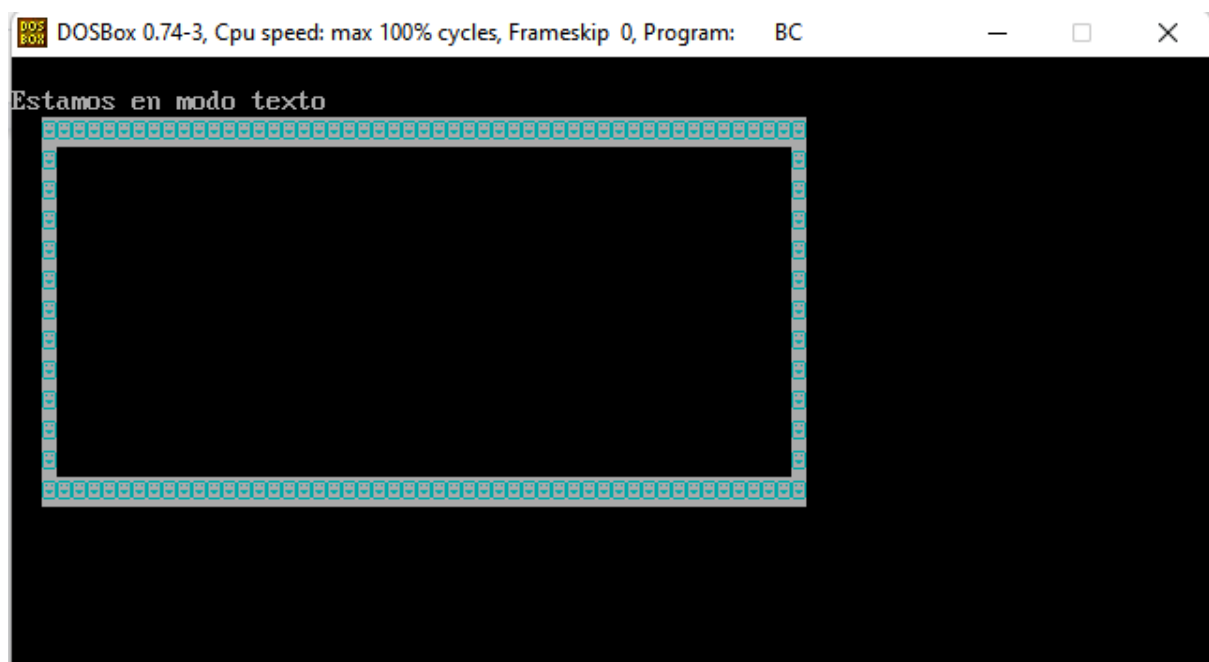
2. REQUISITOS AMPLIADOS

2.1. Implementar una función que permita dibujar un recuadro en la pantalla en modo texto. Recibirá como parámetros las coordenadas superior izquierda e inferior derecha del recuadro, el color de primer plano y el color de fondo.

Para esta función tenemos como parámetros las coordenadas x,y de la parte superior izquierda, las coordenadas de la parte inferior derecha y el carácter que va a mostrar.

Se lee de teclado el color de fondo y el color del carácter que queremos poner, y al igual que anteriormente guardamos estos valores en las variables globales creadas para el fin de cambiar el color del texto.

Posteriormente se hacen unos bucles for que recorren los cuatro lados del recuadro, primero el lado superior, después el lado inferior, sigue el lado izquierdo y por último el lado derecho. Para escribir el carácter con color, se llama a la función de escribir un carácter en pantalla con color.



2.2. Implementar en lenguaje C un programa que establezca modo gráfico CGA (modo=4) para crear dibujos sencillos en pantalla.

Para la implementación de esta función, hemos utilizado llamadas a otras dos funciones auxiliares: la función *pixel* y la función mediante la cual cambiábamos el modo de video.

El dibujo que hemos decidido realizar ha sido una carita sonriente.

Hemos dividido la imagen en varias partes, cada una de ellas implementada mediante un bucle *for*.

En primer lugar, hemos desarrollado las líneas rectas (tanto verticales como horizontales), obteniendo dos de cada tipo. A continuación, buscamos dibujar las líneas horizontales necesarias para unir las anteriormente creadas entre sí. De esta forma, ya tendríamos el perímetro de la cara.

El siguiente paso sería dibujar los ojos. Volvemos a hacer uso, en este caso de un solo bucle, para pintar los dos ojos, siendo dos líneas rectas verticales en la parte interior del perímetro dibujado anteriormente.

Por último, pintamos la sonrisa. Esta sonrisa está formada por tres bucles: dos para realizar dos líneas diagonales de los lados y otro para la línea horizontal que las une.

El punto a tener en cuenta en este dibujo son las coordenadas de cada uno de los puntos, los cuales hemos ido calculando gráficamente con una cuadrícula.

Gracias a la función *pixel*, útil en modo gráfico, hemos podido realizar el dibujo. A esta función se le pasa como parámetros la coordenada del eje X, la coordenada del eje Y, y el color. En nuestro caso, para el contorno de la cara hemos utilizado el color 2 (rosa), mientras que para los rasgos de la cara hemos usado el color 1 (azul).

Vemos una imagen del resultado obtenido:

