PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA

Universidad de Granada

Práctica 1

Entrada/Salida utilizando interrupciones con lenguaje C

Antonio Fernández Ortega

GOTXY()

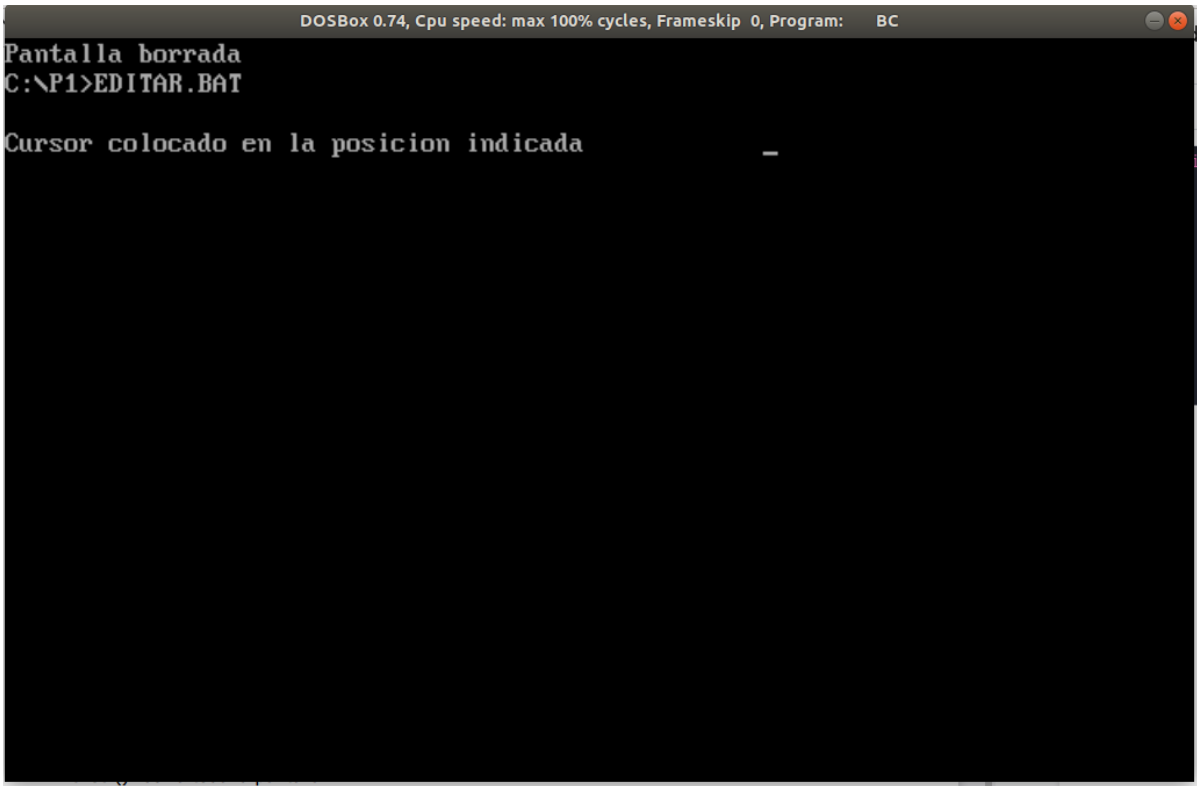
Coloca el cursor en la posición que le indiquemos en los parámetros.

```
// Colocar el cursor en una posición determinada

void gotoxy(int x, int y){
   union REGS inregs, outregs;

   inregs.h.ah=0x02;
   inregs.h.bh=0x00;
   inregs.h.dh=y; // Coordenada y
   inregs.h.dl=x; // Coordenada x

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



SETCURSORTYPE()

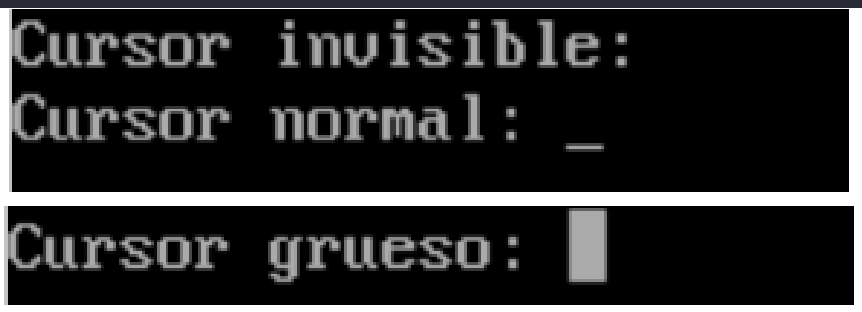
Fija el aspecto del cursor dependiendo del parámetro que se le introduzca a la función. Si es 0 es invisible, si es 1 es normal y si es 2 es grueso.

```
// Fijar el aspecto del cursor

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);
}
```

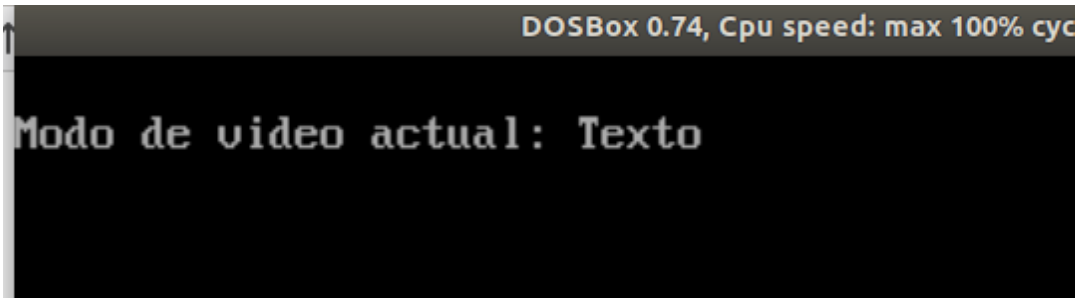


SETVIDEOMODE()
Fija el modo de vídeo que se le pasa como parámetro.

GETVIDEOMODE()
Extrae el modo de vídeo en el que se encuentra tras la llamada a la interrupción 86.
La extrae del parámetro al y luego comprueba en qué modo está.

```
// Fijar el modo de vídeo deseado
void setvideomode(int x){
   union REGS inregs, outregs;
   inregs.h.ah=0x00;
   inregs.h.al=x;
   int86(0x10, &inregs, &outregs);
}

// Obtener el modo de vídeo actual
void getvideomode(){
   union REGS inregs, outregs;
   int modo;
   int numColumnas;
   inregs.h.ah=0xF;
   int86(0x10, &inregs, &outregs);
   modo=outregs.h.al;
   numColumnas=outregs.h.ah;
   if(modo<=3 || modo==7){
       printf("Texto");
   }else{
       printf("Grafico");
   }
}
```



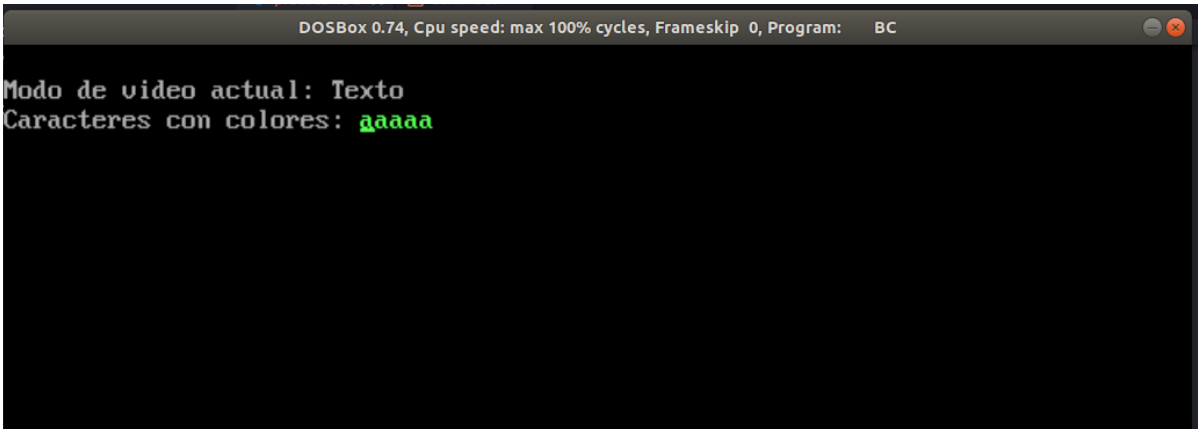
TEXTCOLOR()
Modifica el color de los caracteres que se van a mostrar. Se le introduce un parámetro que es un número que indica el color del texto. El parámetro al es la letra que va a mostra, en este caso el 97 es la letra ‘a’ y el parámetro cx es el número de veces que la va a mostrar, que en este caso lo muestra 5 veces.

```
// Modificar el color de primer plano con el que se mostrarán los
caracteres

void textcolor(int CTexto){
   union REGS inregs, outregs;

   inregs.h.ah=0x09;
   inregs.h.al=97;
   inregs.h.bl=CTexto;
   inregs.h.bh=0x00;
   inregs.x.cx=5;

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



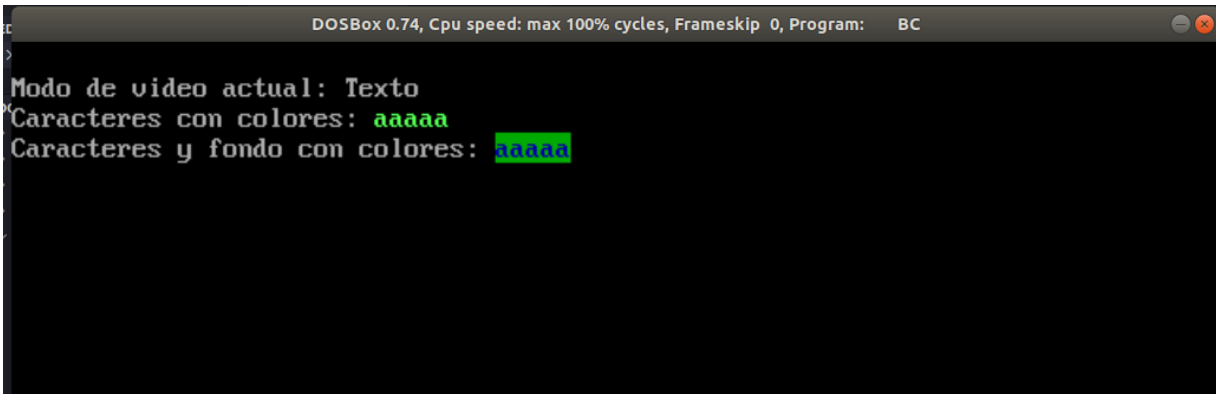
TEXTBACKGROUND()
Funciona exactamente igual que el método anterior solo que se le añade un nuevo parámetro que es para indicar el número del color que deseamos para el fondeo de los caracteres.

```
// Modificar el color de fondo con el que se mostrarán los caracteres

void textbackground(int CTexto, int CFondo){
   union REGS inregs, outregs;
   int color=CFondo << 4 | CTexto;

   inregs.h.ah=0x09;
   inregs.h.al=97;
   inregs.h.bl=color;
   inregs.h.bh=0x00;
   inregs.x.cx=5;

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

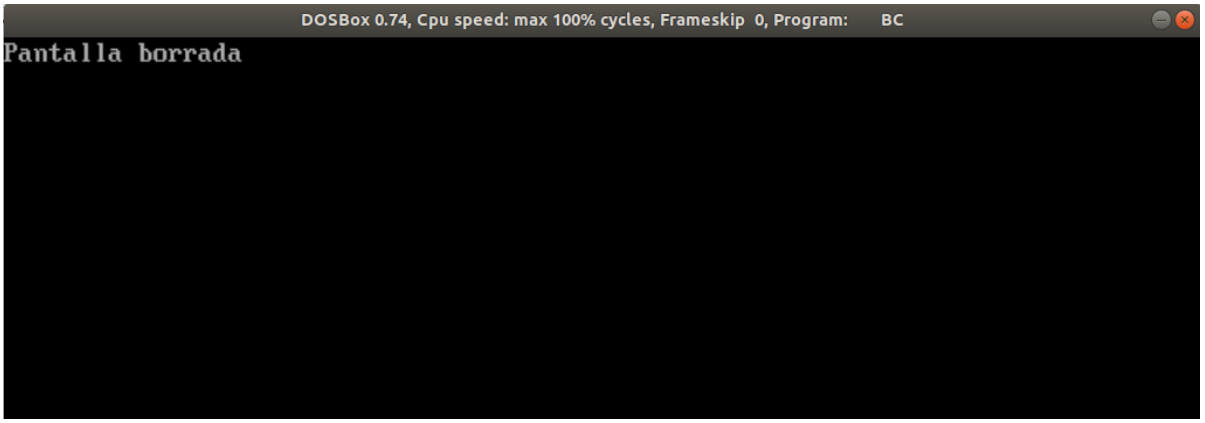


CLRSCR()
Borra todo el contenido de la pantalla.

```
// Borrar toda la pantalla

void clrscr(){
   union REGS inregs, outregs;

   inregs.h.ah=0x15;
   int86(0x10, &inregs, &outregs);
   inregs.h.ah=0x00;
   int86(0x10, &inregs, &outregs);
}
```



CPUTCHAR()
Recibe como parámetro un carácter y posteriormente lo muestra.

```
void cputchar(char c){
   union REGS inregs, outregs;

   inregs.h.ah=2;
   inregs.h.dl=c;

   int86(0x21, &inregs, &outregs);
}
```

GETCHE()
Devuelve el carácter recibido por el teclado.

```
// Obtiene un carácter de teclado y lo muestra en pantalla

int getche(){
   union REGS inregs, outregs;
   int caracter;

   inregs.h.ah=0x01;

   int86(0x21, &inregs, &outregs);

   caracter=outregs.h.al; // Código ASCII del carácter
   return caracter;
}
```

DIBUJARCUADRADO()
Dibuja un cuadrado teniendo como parámetros las coordenadas de las esquinas y el color del cuadrado.

```
// Función para dibujar un cuadrado

void dibujarCuadrado(int fila1, int col1, int fila2, int col2, int
ctexto, int cfondo){
   union REGS inregs, outregs;

   inregs.h.ah=0x06;
   inregs.h.al=0;
   inregs.h.bh=cfondo << 4 | ctexto;
   inregs.h.ch=fila1;
   inregs.h.cl=col1;
   inregs.h.dh=fila2;
   inregs.h.dl=col2;

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