PDIH

Periféricos y Dispositivos de Interfaz Humana.

Práctica 1: Entrada/Salida utilizando interrupciones con lenguaje C.

Autora: Cristina María Crespo Arco
Correo: cmcrespo@correo.ugr.es
Autor: Andrés Piqueras Brück
Correo: andrespiqueras@correo.ugr.es

Profesor: Pedro A. Castillo Valdivieso

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Curso 2021 - 2022

Índice

Para la realización de esta prácticas hemos implementado las siguientes funciones:

1. gotoxy(int x, int y)

Pasamos las coordenadas a las que irá el cursor, y ponemos las variables tal y como se especifican en el apartado del guión "Colocar el cursor en una posición determinada".

1.1. Código:

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

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

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

2. setcursortype(int tipo)

"Robado" directamente del ejemplo propuesto. Dependiendo del entero introducido como parámetro, cambian los valores de inregs.h.ch y inregs.h.cl, cambiando así el tipo de cursor.

2.1. Código:

```
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);
}
```

2.2. Ejecución:

Figura 1: Ejecución función setcursortype(int tipo).

3. setvideomode(int x)

Pasamos un entero que tiene que ir en hexadecimal, y cambiamos el modo de vídeo, tal y como se especifica en el guión "Seleccionar el modo de vídeo".

3.1. Código:

```
void setvideomode(int x){
    union REGS inregs, outregs;
    inregs.h.ah=0;
    inregs.h.al=x; // Resolución: 80x25 / Colores: 16 / Tipo: Texto

    printf("\n%d\n", x);
    int86(0x10, &inregs, &outregs);
}
```

4. getvideomode()

Hecho tal y como se especifica en el guión "Averiguar el modo de vídeo actual". Además, según el valor que se obtenga, se devuelve en printf el modo de vídeo en el que nos encontramos. 0X03 indica que estamos en modo texto, 0x04 indica que estamos en modo gráfico, entre otros.

4.1. Código:

```
void getvideomode(){
    union REGS inregs, outregs;

    inregs.h.ah=0xF;
    inregs.h.al=x;

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

    x=outregs.h.al;

    if (x == 0x03){
        printf("\nModo texto");
    }else if (x == 0x02){
        printf("\nModo samba");
    }else if (x == 0x01){
        printf("\nmodo epico");
    }else if (x== 0x04){
        printf("\nmodo grafico");
    }else{
        printf("\nmodo raro");
    }
}
```

5. textcolor(int color)

Pasamos el color como parámetro, y es el valor que le daremos a inregs.h.bl, que es el color de la letra. El resto de valores los ponemos como se indica en el guión, y inregs.h.al lo ponemos al gusto, ya que es el campo que indica el código ascii del carácter que vayamos a escribir.

5.1. Código:

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

    inregs.h.ah=0x09;
    inregs.h.al=98;
    inregs.h.bl=color;
    inregs.h.bh=0x00;
    inregs.x.cx=10;

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

5.2. Ejecución:

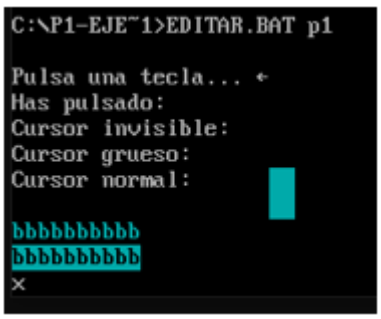

Figura 2: Ejecución función textcolor(int color).

6. textbackground(int color)

Similar al apartado anterior, pero esta vez cambiamos el primer cuarteto, para cambial el color del fondo.

6.1. Código:

```
void textbackground(int color){
    union REGS inregs, outregs;

    inregs.h.ah=0x09;
    inregs.h.al=98;
    inregs.h.bl=color << 4;
    inregs.h.bh=0x00;
    inregs.x.cx=10;

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

6.2. Ejecución:

Figura 3: Ejecución función textbackground(int color).

7. clrscr()

Imprimimos muchas veces \n.

7.1. Código:

```
void clrscr(){
    printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
            \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
            \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
            \n\n\n");
}
```

7.2. Ejecución:

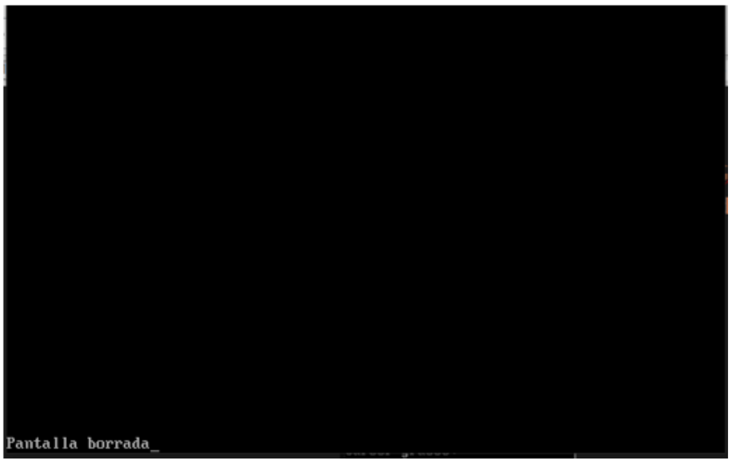

Figura 4: Ejecución función clrscr().

8. cputchar(char c)

Colocamos un carácter, como se indica en el guión.

8.1. Código:

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

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

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

9. getche(int tmp)

Reciclamos las funciones mi_getchar y mi_putchar de los ejemplos.

9.1. Código:

```
getche(int tmp){
    printf("\nPulsa una tecla... ");
    tmp = mi_getchar();

    printf("\nHas pulsado: ");
    mi_putchar( (char)tmp );
}
```

10. DibujarCuadradoModoTexto(int f1, int c1, int f2, int c2, int primerPlano, int fondo)

ch, cl, dh y dl son dos puntos, que definirán la localización del cuadrado. Bh el color de fondo.

10.1. Código:

```
void dibujarCuadradoModoTexto(int f1, int c1, int f2, int c2,
                              int primerPlano, int fondo){
    union REGS inregs, outregs;

    inregs.h.ah=0x06;
    inregs.h.al=0;
    inregs.h.bh=fondo << 4 | primerPlano;
    inregs.h.ch=f1;
    inregs.h.cl=c1;
    inregs.h.dh=f2;
    inregs.h.dl=c2;

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

11. modoGraficoCGA(int x, int y, int color)

Esta función simplemente dibuja un punto en modo gráfico, en las coordenadas especificadas y con el color dado.

11.1. Código:

```
void modoGraficoCGA(int x, int y, int color){
    union REGS inregs, outregs;

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

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

12. caradeCreeper

Llamo en el main a la función modoGraficoCGA en varios bucles, de forma que dibujo varias lineas rectas en en las coordenadas que me interesan, y consigo formar la figura de un Creeper.

12.1. Ejecución:

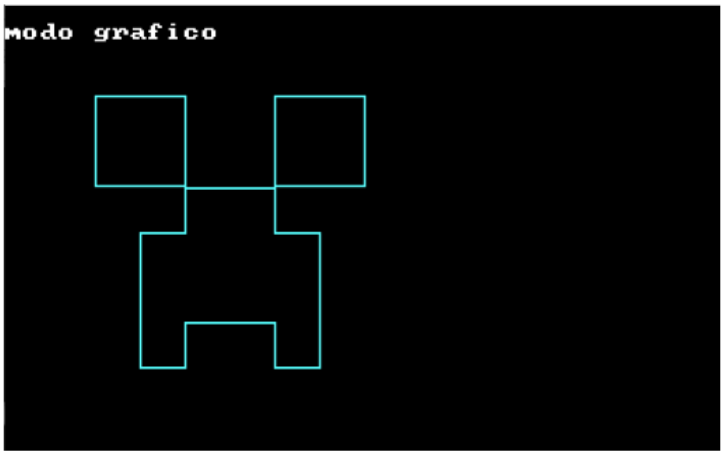

Figura 5: Ejecución caradeCreeper.