



UNIVERSIDAD DE GRANADA

Práctica 1 de Periféricos y Dispositivos de Interfaz Humana

Curso 2024-2025

Francisco Quiles Ramírez

Índice:

gotoxy.....	3
setcursortype.....	3
setvideomode.....	4
getvideomode.....	4
textcolor.....	4
textbackground.....	4
clrscr.....	4
cputchar.....	5
getche.....	5
pixel.....	5
Ejercicio extra 1.....	6
Ejercicio extra 2.....	6
Ejercicio extra 3.....	7
Main y salidas.....	7

Gotoxy

Mueve el cursor a la posición (x,y) en pantalla en modo texto, para ello usa la interrupción 0x10, y función 0x02 para manejo de video , dh representa la fila y dl la columna.

```
//Ej 1
void gotoxy(int x, int y) {
    union REGS inregs, outregs;
    inregs.h.ah = 0x02; // Indicamos
    inregs.h.dh = y; // Fila
    inregs.h.dl = x; // Columna
    inregs.h.bh = 0x00; // (En la ve
    int86(0x10, &inregs, &outregs);
}
```

Setcursortype

Cambia la apariencia del cursor en modo texto, permitiendo hacerlo invisible, normal o grueso, usando la interrupción 0x10 y la función 0x01 que controla el tipo de cursor, para ello cambia ch y cl para definir la forma del cursor.

```
//Ej 2
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

Cambia el modo de video entre modo texto y modo gráfico, utilizando la interrupción 0x10, y función 0x00 que cambia la pantalla del sistema, el parámetro modo se pasa en el registro al para indicar el nuevo modo.

```
//Ej3
void setvideomode(BYTE modo){
    union REGS inregs, outregs;
    inregs.h.al = modo;
    inregs.h.ah = 0x00;
    int86(0x10, &inregs, &outregs);
}
```

Getvideomode

Obtiene el modo de video actual que está en uso en el sistema, usa la interrupción 0x10 y la función 0x0F que devuelve el modo de video en al, si el resultado es 3 estará en modo texto y si es 4 en modo gráfico.

```
//Ei4
int getvideomode() {

    int mode;
    union REGS inregs, outregs;
    inregs.h.ah = 15;
    int86(0x10, &inregs, &outregs);
    mode = outregs.h.ah;
    return mode;
}
```

Textcolor

Cambia el color de texto, almacenando el valor en `tcolor`, haciendo que funciones como `cputchar()` utilicen estos valores para mostrar el texto.

```
// EJ 5
void textcolor(unsigned char color) {
    tcolor = color;
}
```

Textbackground

Cambia el color del fondo, almacenando el valor en `bcolor`, haciendo que funciones como `cputchar()` utilicen estos valores para mostrar el texto.

```
// EJ 6
void textbackground(unsigned char color) {
    bcolor = color;
}
```

Clrscr

Simula la limpieza de pantalla en modo texto imprimiendo múltiples saltos de línea (\n).

```
// EJ 7  
void clrscr() {  
    printf("\n\n\n\n\n\n\n\n\n\n\n\n")  
}
```

Cputchar

Imprime un caracter en pantalla con el color de texto y fondo seleccionado. Usa la interrupcion 0x10 y la funcion 0x09, que imprime caracteres en la pantalla con atributos de color. EL color se define combinando bcolor y tcolor en el registro bl.

```
//Ei 8
void cputchar(char c){
    union REGS inregs, outregs;

    inregs.h.ah = 0x09;
    inregs.h.al = c;
    inregs.h.bl = (bcolor << 4) | tcolor;
    inregs.h.bh = 0x00;
    inregs.x.cx = 1;
    int86(0x10, &inregs, &outregs);
}
```

Getche

Captura un carácter del teclado sin necesidad de presionar Enter. Usa la interrupción 0x21 y la función 0x01 que lee un carácter desde la entrada estándar y lo devuelven en al.

```
//Ej 9
int getche(){
    union REGS inregs, outregs;
    int character;

    inregs.h.ah = 1;
    int86(0x21, &inregs, &outregs);

    character = outregs.h.al;
    return character;
}
```

Pixel

Permite dibujar un pixel individual en la pantalla en modo gráfico. Para hacerlo usa la interrupción 0x10, y la función 0x0C, que coloca un punto en coordenadas x,y con el color c. Las coordenadas se almacenan en cx y dx y el color en al.

```
//Ej 10
void pixel(int x, int y, BYTE C){
    union REGS inregs, outregs;
    inregs.x.cx = x;
    inregs.x.dx = y;
    inregs.h.al = C;
    inregs.h.ah = 0x0C;
    int86(0x10, &inregs, &outregs);
}
```

Ejercicio extra 1

Se ha implementado la función dibujar cuadrado que dibuja un cuadrado en modo texto usando gotoxy() para posicionar cada borde y cputchar() para imprimir caracteres. Primero imprime el borde exterior con un color y luego dibuja un cuadrado más pequeño dentro de otro color.

```

//Ej extra 1 dibujar cuadrado
void dibujarcuadrado(int xsi, int ysi, int xid, int yid, unsigned char text1, unsigned char bg1, unsigned char text2, unsigned char bg2) {

    int i;

    clrscr();
    printf("\nEjercicio extra: Pintar cuadrado");

    textcolor(text1);
    textbackground(bg1);
    //Arriba
    for (i = xsi; i <= xid; ++i) {
        gotoxy(i, ysi);
        cputchar(' ');
    }
    //Abajo
    for (i = xsi; i <= xid; ++i) {
        gotoxy(i, yid);
        cputchar(' ');
    }
    //Izquierda
    for (i = ysi; i <= yid; ++i) {
        gotoxy(ysi, i);
        cputchar(' ');
    }
    //Derecha
    for (i = ysi; i <= yid; ++i) {
        gotoxy(yid, i);
        cputchar(' ');
    }

    textcolor(text2);
    textbackground(bg2);

    // Arriba
    for (i = xsi + 2; i <= xid - 2; ++i) {
        gotoxy(i, ysi + 2);
        cputchar(' ');
    }
    // Abajo
    for (i = xsi + 2; i <= xid - 2; ++i) {
        gotoxy(i, yid - 2);
        cputchar(' ');
    }
    // Izquierda
    for (i = ysi + 2; i <= yid - 2; ++i) {
        gotoxy(xsi + 2, i);
        cputchar(' ');
    }
    // Derecha
    for (i = ysi + 2; i <= yid - 2; ++i) {
        gotoxy(xid - 2, i);
        cputchar(' ');
    }

    gotoxy(xsi, yid + 2);

}

```

Ejercicio extra 2

Dibuja un rectángulo en modo gráfico usando la función pixel. Se traza la línea superior e inferior iterando sobre el eje x y las líneas izquierda y derecha iterando sobre el eje y.

```

//Ejercicio extra 2, modo grafico
void rectangulo(int x1, int y1, int x2, int y2, BYTE C){
    int i;
    printf("\nEjercicio extra2: Cuadrado");

    // Línea superior e inferior
    for (i = x1; i <= x2; i++) {
        pixel(i, y1, C);
        pixel(i, y2, C);
    }
    // Línea izquierda y derecha
    for (i = y1; i <= y2; i++) {
        pixel(x1, i, C);
        pixel(x2, i, C);
    }
}

```

Ejercicio extra 3

Esta función crea tres patos en ASCII Art en modo texto, para asegurarse de que el dibujo sea preciso se ha usado gotoxy() para color cada carácter en su posición y printf para poner los caracteres

```
void dibujarPatos(){
    //patol
    gotoxy(10, 5); printf("_");
    gotoxy(11, 5); printf("_");
    gotoxy(12, 5); printf("(");
    gotoxy(13, 5); printf(".");
    gotoxy(14, 5); printf(")");
    gotoxy(15, 5); printf("<");

    gotoxy(13, 4); printf("_");

    gotoxy(10, 6); printf("\n");
    gotoxy(11, 6); printf("_");
    gotoxy(12, 6); printf("_");
    gotoxy(13, 6); printf("_");
    gotoxy(14, 6); printf(")");

    //pato2
    gotoxy(17, 5); printf("_");
    gotoxy(18, 5); printf("_");
    gotoxy(19, 5); printf("(");
    gotoxy(20, 5); printf(".");
    gotoxy(21, 5); printf(")");
    gotoxy(22, 5); printf(">");

    gotoxy(20, 4); printf("_");

    gotoxy(17, 6); printf("\n");
    gotoxy(18, 6); printf("_");
    gotoxy(19, 6); printf("_");
    gotoxy(20, 6); printf("_");
    gotoxy(21, 6); printf(")");
    gotoxy(22, 6); printf(">");

    //pato3
    gotoxy(24, 5); printf("_");
    gotoxy(25, 5); printf("_");
    gotoxy(26, 5); printf("(");
    gotoxy(27, 5); printf(".");
    gotoxy(28, 5); printf(")");
    gotoxy(29, 5); printf("=");

    gotoxy(27, 4); printf("_");

    gotoxy(24, 6); printf("\n");
    gotoxy(25, 6); printf("_");
    gotoxy(26, 6); printf("_");
    gotoxy(27, 6); printf("_");
    gotoxy(28, 6); printf(")");
    gotoxy(29, 6); printf("=");
}
```

Main y salidas

Aquí probaremos tanto las 10 funciones que se piden más las extra, probando algunas juntas y otras por separado.

```
int main(){
    int tmp;
    int video_mode;
    clrscr();
    printf("\nEjercicio 7: Se ha limpiado la pantalla.");

    //Ej 1 Gotoxy
    printf("\nEjercicio 1: Pulsa una tecla (GOTOXY 20,20): ");
    gotoxy(20,20);
    mi_pausa();
    printf("\n");

    //Ej 2
    printf("\nEjercicio 2: Cursor invisible: ");
    setcursortype(0);
    mi_pausa();
    printf("\nEjercicio 2: Cursor grueso: ");
    setcursortype(2);
    mi_pausa();
    printf("\nEjercicio 2: Cursor normal: ");
    setcursortype(1);
    mi_pausa();

    //Ej 3 y 4
    printf("\nEjercicio 3 y 4: Pulsa una tecla (setVideo): ");
    setvideomode(MODOGRAFICO);
    video_mode = getvideomode();
    printf("\nEjercicio 3 y 4: Pulsa una tecla (MODO GRAFICO 4): %i", video_mode);
    mi_pausa();
    printf("\nEjercicio 3 y 4: Pulsa una tecla (MODO TEXTO 3): ");
    setvideomode(MODOTEXTO);
    video_mode = getvideomode();
    printf("\nEjercicio 3 y 4: (MODO TEXTO 3) video mode: %i", video_mode);
    mi_pausa();
    printf("\n");

    //Ej 5, 6, 8 y 9
    printf("\nEjercicios 5, 6, 8 y 9: Introduce una tecla: (GETCHE)");
    tmp = getche();
    mi_pausa();
    printf("\nEjercicios 5, 6, 8 y 9: Pulsa una tecla: ( TEXTCOLOR 1 (azul oscuro) y background 3 (cian) )\n");
    mi_pausa();
    textcolor(1);
    textbackground(3);
    cputchar((char)tmp);
    mi_pausa();
    printf("\n");
}
```

```

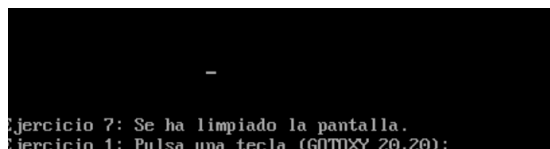
//Ej 10
printf("\nEjercicio 10: Pixel: ");
setvideomode(MODOGRAFICO); //gráfico
pixel(10,40,0);
pixel(10,50,1);
pixel(15,60,2);
pixel(20,70,3);
mi_pausa();
setvideomode(MODOTEXTO); //texto

//Ej extra1
mi_pausa();
dibujarcuadrado(1,1,10,10,1,3,14,4);
mi_pausa();
printf("\n");
//Ej extra2
setvideomode(MODOGRAFICO);
rectangulo(80,80,180,140,3);
mi_pausa();
setvideomode(MODOTEXTO);
//Ej extra3
printf("\nEjercicio extra3: ");
dibujarPatos();

return 0;
}

```

En primer lugar se ha limpiado la pantalla con clrscr y aparece gotoxy que nos lleva el cursor a la posición (20,20)

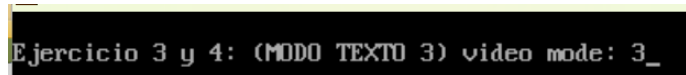


Ahora se ha llamado a setcursortype de las tres formas posibles

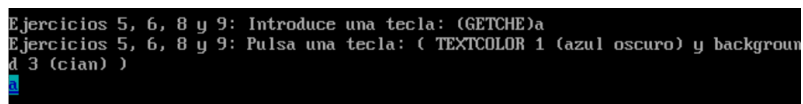




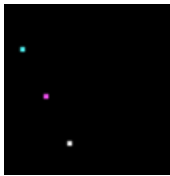
Tras una pausa entramos en modo gráfico y después en modo texto con setvideomode. Debajo con getvideomode vemos que nos dice que está en el modo texto que es el 3

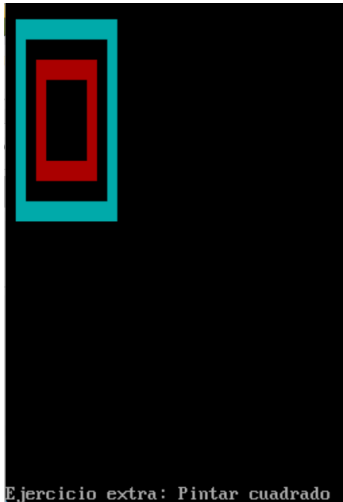
Aquí hemos combinado la funcionalidad de cputchar y getch para tomar una letra y cambiarle el color y fondo con textcolor y textbackground a un tono azul con el fondo cian.



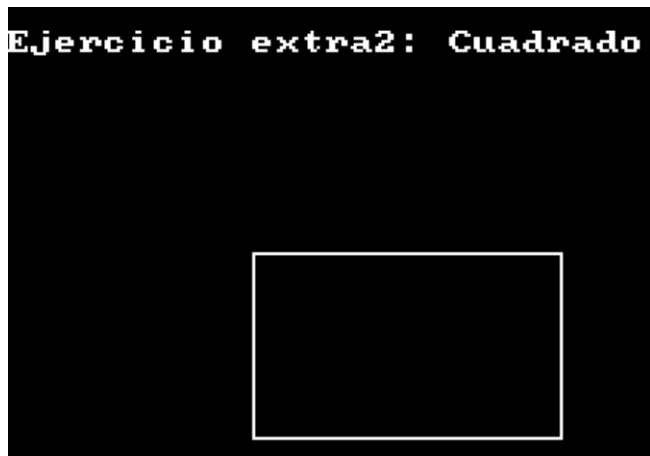
Ahora se ha usado pixel y se han representado tres puntos en el espacio en modo gráfico.



Como primer ejercicio extra se han hecho dos cuadrados en modo texto, uno dentro de otro



En este se ha hecho un cuadrado en modo gráfico.



Finalmente para el último ejercicio extra podemos ver un ASCII Art con tres patos.

