

Práctica 1



**UNIVERSIDAD
DE GRANADA**

Antes de comenzar con las funciones que se requerían, he hecho 3 de uso general para el programa.

[illegible]

Tanto *mi_pausa()* como *mi_exit()* utilizan la interrupción de 21h de MSDOS. La primera pausa el programa hasta detectar la pulsación de una tecla y la segunda es el equivalente al código *return 0* para el main del programa.

Clrscr únicamente imprime por pantalla muchos retornos de carro para simular el comando *cls* de la terminal.

En el main del programa simplemente se van ejecutando las funciones para comprobar su correcto funcionamiento.

```
int main(){
    int mode;
    clrscr();
    printf("\nGotoxy: Coloca el cursor en una posición determinada");

    gotoxy(20,20);
    printf("xx");
    printf("\nPulsa una tecla para continuar");
    mi_pausa();
    clrscr();
}
```

```
printf("\nSetcursortype: Fija el aspecto del cursor");

printf("\nCursor invisible: ");
setcursortype(0);
mi_pausa();
printf("\nCursor grueso: ");
setcursortype(2);
mi_pausa();
printf("\nCursor normal: ");
setcursortype(1);
mi_pausa();

printf("\nPulsa una tecla para continuar");
mi_pausa();

printf("\nSetvideomode: Fijar el modo de video");
setvideomode(4); //Modo gráfico de 4 colores
pixel(10,40,0);
pixel(15,50,1);
pixel(20,60,2);
pixel(25,70,3);

mi_pausa();
setvideomode(3); //Modo texto 16 colores

printf("\nPulsa una tecla para continuar");
mi_pausa();
```

```
printf("\nGetvideomode: Obtener el modo de video actual");
mode = getvideomode();
printf("\nModo de video %i", mode);
mi_pausa();

printf("\nPulsa una tecla para continuar");
mi_pausa();

printf("\nGetch: Se captura una tecla y se imprime por pantalla");
getch();

printf("\nPulsa una tecla para continuar");
mi_pausa();

printf("\nTextcolor, background color y cputchar: Modifica color de primer plano,
textcolor(1);
backgroundcolor(4);
cputchar('A');

mi_pausa();

textcolor(128);
backgroundcolor(2);
cputchar('A');

mi_pausa();
asciiart();
mi_pausa();

mi_exit();
}
```

Para el resto de funciones excepto para *getch()* se utiliza la interrupción BIOS de vídeo 10h. La primera función es *gotoxy()*. En el registro AH se indica el número de función y en los registros DL y DH las coordenadas donde se quiere colocar el cursor. Para comprobarlo se imprime por pantalla XX.

```
void gotoxy(int x, int y){
    union REGS inregs, outregs;
    inregs.h.ah = 0x02;
    inregs.h.bh = 0x00;
    inregs.h.dh = y;
    inregs.h.dl = x;
    int86(0x10, &inregs, &outregs);
    return;
}
```

```
XX
Pulsa una tecla para continuar_
Gotoxy: Coloca el cursor en una posición determinada
```

Setcursortype() fija el tamaño del cursor en modo texto. En AH se introduce el número de función 1 y en CH y CL el número de línea inicial y el número de línea final respectivamente. En la función se definen 3 tipos de cursor para que sea mas sencillo ejecutarlo.

```
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);
}
```

```
Setcursortype: Fija el aspecto del cursor
Cursor invisible:
Cursor grueso: █
```

```
Setcursortype: Fija el aspecto del cursor
Cursor invisible:
Cursor grueso:
Cursor normal: _
```

Setvideomode() define el modo de video, en AH se indica el número de función 0 y en AL el modo de video a establecer. Para complementar el funcionamiento he creado una función *pixel()* que pinta un pixel de un color determinado. El número de función es Ch, en los registros CX y DX se indican las coordenadas del pixel y en AL el color. El primer pixel en la diagonal no se puede ver ya que el color es 0 (negro).

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

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);
}
```



Getvideomode() simplemente averigua el modo de video que se esta utilizando. En el registro AH se indica el número de función Fh y en el registro AL se almacena el modo de video.

```
void getch(){
    union REGS inregs, outregs;
    char character;

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

    character = outregs.h.al;
    printf("\nSe ha pulsado: %c", character);
}
```

```
Getvideomode: Obtener el modo de video actual
Modo de video 3_
```

La función *getch()* utiliza la interrupción 21h de MSDOS y capta por teclado un carácter y lo muestra por pantalla. Utiliza el número de función 1 y el carácter capturado se almacena en el registro AL.

```
void getch(){
    union REGS inregs, outregs;
    char character;

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

    character = outregs.h.al;
    printf("\nSe ha pulsado: %c", character);
}
```

```
Getch: Se captura una tecla y se imprime por pantalla
Se ha pulsado: k
Pulsa una tecla para continuar
```


Para hacer funcionar *textcolor()*, *backgroundcolor()* y *cputchar()* se crean unas variables que serán modificadas por las dos primeras funciones. En *cputchar()* se utiliza el número de función 9, en el registro AL se introduce el carácter a imprimir por pantalla, en CX el número de repeticiones y en BL se accede a las variables mencionadas y se introducen en el registro.

```
void textcolor(BYTE color){
    txtcolor = color;
}

void backgroundcolor(BYTE color){
    bgcolor = color;
}

void cputchar(char c){
    union REGS inregs, outregs;
    inregs.h.ah = 0x09;
    inregs.h.al = c;
    inregs.h.bl = bgcolor << 4 | txtcolor;
    inregs.h.bh = 0x00;
    inregs.x.cx = 1;
    int86(0x10,&inregs,&outregs);
}
```

Textcolor, background color y cputchar: Modifica color de primer plano, fondo y poner un caracter

Por último la función *asciiart()* utiliza *gotoxy()* para pintar un dibujo ASCII simple por pantalla.

```
void asciiart(){
    clrscr();
    gotoxy(21,10);
    printf("\\_/_");
    gotoxy(20,11);
    printf("(=^.^=)");
    gotoxy(19,12);
    printf("''_)_(''");
}
```

