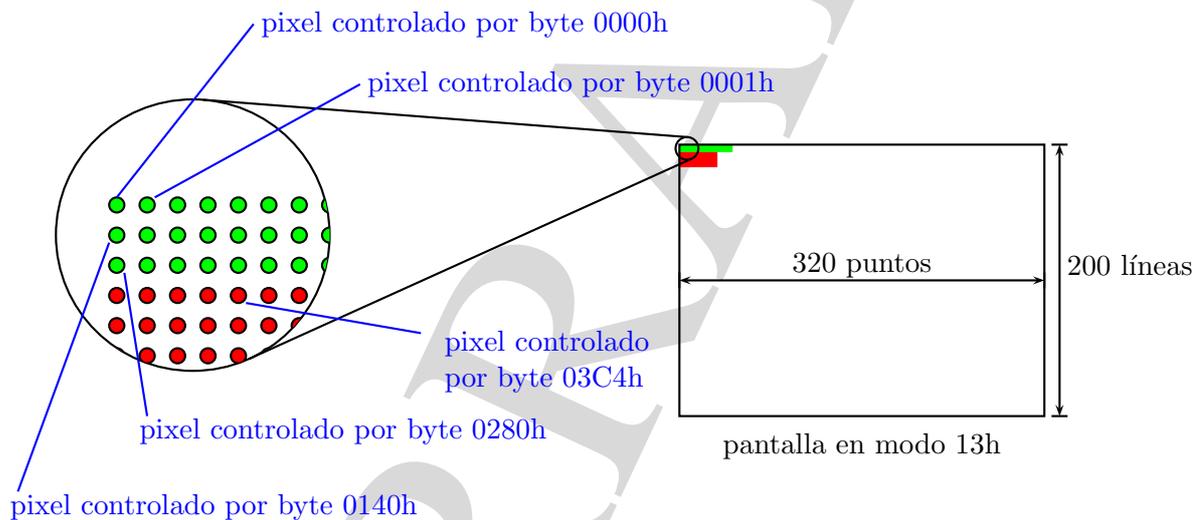


Ejercicio TC72-34p

Escribir una rutina en lenguaje ensamblador del 80x86 para pintar un punto en el modo 13h del adaptador VGA (320x200, 256 colores). Las coordenadas X-Y del punto y el color se les pasan a la rutina en los registros AX, BX y CX, respectivamente.

Solución

El modo 13h del adaptador gráfico VGA proporciona una resolución de 320x200 y 256 colores simultáneos. La programación de este modo es más simple que los modos de 16 colores del VGA ya que aquí la memoria se presenta como un único espacio (no hay planos) de 64000 bytes, que comienza en A000:0000, y donde cada byte controla un pixel. El byte en desplazamiento 0 controla el pixel superior izquierda de la pantalla, el byte en el desplazamiento 319 controla el pixel de la esquina superior derecha de la pantalla, el byte en el desplazamiento 320 controla el segundo pixel hacia abajo en la izquierda de la pantalla, y el byte en la posición de desplazamiento 63999 controla el pixel de la esquina inferior derecha de la pantalla.



Vista la organización de la memoria de pantalla, para pintar un pixel en las coordenadas (X,Y) de la pantalla lo que hay que determinar es qué byte (en qué desplazamiento está) controla dicho pixel y escribir ahí el color deseado, ya que un byte puede tener 256 valores diferentes.

La fórmula para el desplazamiento es:

$$\text{OFFSET} = 320 \times Y + X$$

y lo único a controlar es que $0 \leq X \leq 319$ y que $0 \leq Y \leq 199$ para no salirse de rango.

```
1 pixel13h proc near
2     push dx
3     push di
4     push es
5     push ax
6     cmp ax,320 ; comprueba que
7     jb @@1 ; 0 <= X <= 319
```

```

8      jmp     @@9
9      @@1:
10     cmp     bx,200           ; comprueba que
11     jb     @@2           ; 0 <= Y <= 199
12     jmp     @@9
13     @@2:
14     mov     di,ax
15     mov     ax,320
16     mul     bx           ; DX:AX = 320*Y
17     add     di,ax       ; OFFSET = 320*Y+X
18     mov     ax,0A0000h  ; segmento pantalla
19     mov     es,ax
20     mov     es:[di],cl  ; se escribe el byte del color
21     @@9:
22     pop     ax
23     pop     es
24     pop     di
25     pop     dx
26     ret
27     pixel13h   endp

```

Con el MASM versión 6 en adelante se podría haber escrito así:

```

      .IF (AX < 320) && (BX < 200)
      mov     di,ax
      mov     ax,320
      mul     bx
      add     di,ax
      mov     ax,0A0000h
      mov     es,ax
      mov     es:[di],cl
      .ENDIF

```