

centro de la pantalla: mov DH,0C y DL,29 o DX,0C29

La pantalla del monitor en la PC x86 está dividida en 80 columnas y 25 filas en modo de texto normal.

- Las columnas están numeradas del 0 al 79.
- Las filas están numeradas del 0 al 24.

A la esquina superior izquierda se le ha asignado 00,00 y a la superior derecha 00,79. Abajo a la izquierda son 24,00, abajo a la derecha 24,79.

Decimal	Hexadecimal
00,00 00,79	00,00 00,4F
centro de pantalla 12,39	centro de pantalla 0C,27
24,00 24,79	18,00 18,4F

Para borrar la pantalla usando INT 10H:

```
mov ah,06 establecer la función de desplazamiento
mov al,00 desplazar toda la pantalla
mov bh,07 con formato estándar
mov ch,00 establecer el valor inicial de la fila
mov cl,00 establecer el valor inicial de la columna
mov dh,18 indicar el valor final de la fila
mov dl,4f indicar el valor final de la columna
int 10 se invoca la interrupción
mov ah,4c
int 21
```

El modo CGA es el denominador común para todos los monitores en color, ya que todos los monitores en color y los circuitos de vídeo son compatibles con versiones posteriores.

byte: PRGBIRGB (P parpadeo RGB q es color de fondo I intensidad RGB q es color de frente)

El fondo puede tomar ocho colores combinando los colores rojo, azul y verde.

El primer plano puede tener cualquiera de los 16 colores combinando rojo, azul, verde e intensidad.

La tarjeta gráfica es un componente que procesa datos relacionados con el vídeo y las imágenes.

Puede venir integrada o ser una unidad externa, se conecta para apoyar al CPU. Estas convierten los datos digitales en imágenes visibles en el monitor. Se componen de una GPU que tiene numerosos núcleos de baja frecuencia, enfocados en el procesamiento de vértices y píxeles.

Además, cuentan con memoria gráfica (GRAM) para almacenar y transportar datos, que si es insuficiente, puede limitar su potencia.

El RAMDAC es un convertidor que transforma las señales digitales generadas por la tarjeta gráfica en señales analógicas, permitiendo que el monitor las interprete. Su capacidad influye en la adaptación a distintas velocidades de refresco. Las gráficas suelen incorporar ventiladores.

El funcionamiento de una gráfica incluye dos funciones clave: el procesamiento de vértices que organiza la posición y rotación de los vértices definiendo qué partes serán visibles en la pantalla y el procesamiento de píxeles que genera las texturas y efectos necesarios para crear gráficos realistas.

Una vez procesados, la gráfica envía los datos al monitor a través de distintos tipos de salidas.

Una GPU integrada está en el procesador o la placa base, comparte la memoria RAM del sistema para procesar gráficos intensivos, lo cual afecta el rendimiento general.

La VRAM es la memoria de vídeo de las tarjetas gráficas, diseñada específicamente para manejar gráficos y videojuegos.

Hay cuatro tipos principales de salidas de vídeo que podemos encontrar en una tarjeta gráfica: VGA, DVI, HDMI y DisplayPort.

VGA transmite señales analógicas RGBHV y no incluye audio, permite conexiones en caliente que puede romper todo por subas de tensión. Para conectar VGA a monitores digitales, se requiere un adaptador activo que convierta la señal analógica a digital.

DVI son parte analógica y parte digital, no transmite audio, su cuerpo principal lleva las señales digitales de la gráfica al monitor, mientras que en el lado derecho están los pines destinados a transmitir la señal analógica.

HDMI no es capaz de transportar señal de vídeo analógica, es capaz de transmitir audio, admite mayores resoluciones y tasas de refresco vertical de la pantalla.

DisplayPort transmite imagen y sonido.

La resolución de un dispositivo es el número de píxeles que éste es capaz de mostrar, expresada en función de su anchura y su altura. El ratio de aspecto / relación de aspecto número q se usa para describir la relación que hay entre la anchura y la altura de la pantalla (ej 16:9)

El teclado funciona como una matriz de cables cruzados, cada intersección tiene un pulsador que conecta una línea horizontal con una vertical. El procesador del teclado envía corriente a una horizontal y sondea las verticales, esto se repite para asegurar que no se pierdan pulsaciones. Cada tecla de un teclado tiene un código único de 7 bits "scan code", permitiendo hasta 128 teclas. Cuando se presiona una tecla, el procesador envía su scan code y, al liberarla, añade 80h y lo envía de nuevo. Esto genera dos bytes de información por pulsación: uno para la presión y otro para la liberación. La BIOS proporciona soporte para el teclado, utilizando un buffer circular que almacena las teclas pulsadas y el estado de teclas, buffer por defecto de 16 teclas (32 bytes).

La interrupción de teclado IRQ 1 (INT 9) se activa cada vez que el controlador de teclado tiene datos disponibles. La rutina de atención a esta interrupción guarda el scan code y el ASCII de cada tecla pulsada, actualizando también el estado del teclado y otros datos del BIOS, actualiza bytes de estado para teclas especiales, y maneja combinaciones como Ctrl-Alt-Del. Además, se pueden modificar las funciones de las teclas pulsadas.

La interrupción BIOS INT 16 permite a las aplicaciones acceder a los códigos de las teclas. Esta interrupción tiene múltiples servicios, como leer el estado y vaciar el buffer, especialmente para teclados de más de 100 teclas.

Interrupción del 8086: luego de cada ciclo de instrucción el proce verifica interrupciones, si hay y quiere reconocerla y atenderla, sigue estos pasos:

1. registro de bandera se coloca en la pila.
2. el indicador de interrupción está deshabilitado (IF = 0).
3. el indicador de trampa está deshabilitado (TF = 0).
4. el registro CS se coloca en la pila.
5. el registro IP se coloca en la pila.
6. se transfiere el control al lugar en el que se realice la correspondiente "Interrupción del Servicio". Se almacena la rutina' (ISR). En efecto, esto sería un gran salto.
7. se ejecuta el programa correspondiente al ISR. La última instrucción en el ISR será IRET.
8. luego, la IP se extrae de la pila.
9. CS sale de la pila.
10. el registro Bandera se extrae de la pila.
11. el control vuelve al punto en el que lo había dejado.

Los pasos del 1 al 6 son las acciones antes de llegar a la rutina del servicio de interrupción, del 8 al 10 ocurren al regresar de la interrupción.

Una rutina de servicio de interrupción (ISR) es un programa que se ejecuta cuando ocurre una interrupción, interrumpiendo la tarea actual del procesador para atender una solicitud específica.

Cada ISR está asociado a un tipo de interrupción y se localiza en la memoria mediante un vector de interrupción, que contiene la dirección de la rutina en formato CS. Cuando se genera una interrupción, el procesador utiliza el número de tipo de la interrupción para localizar su vector correspondiente, carga los valores de CS e IP, y comienza a ejecutar la ISR adecuada.

ejemplo: Encuentre la dirección (en el IVT) del vector de interrupción de INT 61H. Encuentre la dirección física del ISR correspondiente a esta interrupción si el vector es 0F00:9872.

solucion: El número de tipo de interrupción es 61H = 97 en decimal. La dirección del vector de interrupción es $97 \times 4 = 388 = 184H$. Por lo tanto, el vector de interrupción debe almacenarse en el IVT

en la ubicación 0000:0184 en adelante. Para ISR, el valor de CS es 0F00H y el valor de IP es 9872H. Se muestra la parte de la tabla de vectores de interrupción que tiene estos vectores almacenados.

<p>escribe en diagonal del monitor y en modo de video 80 x 25 el carácter *</p> <pre> mov cx,0019 mov dh,00 mov dl,00 * mov bh,00 mov ah,02 int 10 mov ah,0a mov bh,00 mov al,2a mov si,cx mov cx,0001 int 10 mov cx,si inc dh inc dl loop * mov ah,4c int 21 </pre>	<p>cargue en el registro BL un número en hexadecimal y que el mismo se muestre en pantalla según su representación binaria</p> <pre> mov cx,0008 recorrer un byte mov ah,02 mov bl,37 mostrar en binario * mov dl,30 mov al,00 shl bl,1 adc dl,al int 21 loop * mov ah,4c int 21 </pre>	<p>escriba 340213-AC en video inverso y en la diagonal del monitor</p> <pre> mov cx,0009 mov di,0300 direc de memoria cn el texto mov dh,00 mov dl,00 * mov bh,00 mov ah,02 int 10 mov ah,09 mov bh,00 mov al,[di] mov si,cx mov bl,70 video inverso int 10 mov cx,si inc dh inc dl inc di loop * mov ah,4c int 21 </pre>
<p>que escriba en el centro de la pantalla el año actual pero con un recuadro de líneas simples. Ejemplo de lo solicitado <u>20XX</u></p> <pre> mov ah,02 mov dh,0c mov dl,29 mov bh,00 int 10 mov ah,02 mov dl,da int 21 mov cx,0004 * mov ah,02 mov dl,c4 int 21 loop * mov ah,02 mov dl,bf int 21 ... </pre>	<p>Escriba un programa que permita ingresar pulsaciones de teclas desde el teclado de la PC y mostrar los caracteres en el monitor del sistema. Pulsando cualquiera de las teclas de función F1-F10 deberían hacer que el programa finalice.</p> <pre> MOV DX,0700 MOV AH,09 INT 21 **MOV AH,00 INT 16 to check the keyboard CMP AL,00 JZ * chequea valor de teclado ingresado MOV AH,0E INT 10 JMP ** * MOV AH,4C INT 21 </pre> <p>[0700] "pulse cualquier letra y F1 hasta F10 para salir del programa"</p>	<p>escriba por pantalla el estado del teclado pero en representación binaria. Utilice el registro BL para cargar el estado del teclado.</p> <pre> mov bx,0000 mov ds,bx mov bl,[417] estado del teclado mov cx,0008 * mov ah,02 mov dl,30 mov al,00 shl bl,1 adc dl,al int 21 loop * mov ah,4c int 21 </pre>
<p>ingresar una clave de tres letras, si la clave no es la correcta que termine el programa, y si es correcta que escriba en centro de la pantalla LOGIN en modo de video 40 x 25 y termine.</p> <pre> mov cx,0003 mov bx,0200 ; direc mem cn la clave mov ah,08 * int 21 mov dl,[bx] cmp dl,al jnz ** inc bx loop * mov ah,00 mov al,01 int 10 </pre>	<pre> mov ah,02 mov dh,0A mov dl,0D mov bh,00 int 10 mov ah,09 mov dx,0203 ; direc mem cn texto "LOGIN\$" int 21 ** mov ah,4c int 21 </pre>	

Realice un programa que lea un carácter del teclado con la interrupción 16h y que luego sea visualizado en pantalla con la interrupción 21h.

```
mov ah,00
int 16
mov ah,02
mov dl,al
int 21
mov ah,4c
int 21
```

obtenga el estado actual de las teclas shift y que luego compruebe el estado de la pulsación de la tecla bloqueo mayúsc, en caso de estar presionada/activada la tecla se debe mostrar por pantalla la letra A, en caso contrario de debe mostrar la tecla d.

```
mov ah,02
int 16
mov di,0200
mov [di],al
and al,40
cmp al,40 ;
jnz *
mov ah,02
mov dl,41
int 21
jmp **
* mov ah,02
mov dl,64
int 21
** mov ah,4c
int 21
```

detecte la pulsación de una tecla y que verifique que sea extendida (Inicio, F1, entre otras), en caso de ser extendida se debe mostrar una cadena de caracteres con el siguiente texto "tecla extendida". El texto a mostrar se debe cargar al comienzo del programa a partir de la posición de memoria 0200h.

```
mov si,0200
mov dl,74
mov [si],dl
inc si
mov dl,65
mov [si],dl
inc si
mov dl,63
mov [si],dl
inc si
mov dl,6c
mov [si],dl
inc si
mov dl,61
mov [si],dl
inc si
mov dl,20
mov [si],dl
inc si
mov dl,65
mov [si],dl
inc si
mov dl,78
mov [si],dl
inc si
mov dl,74
mov [si],dl
inc si
mov dl,65
mov [si],dl
inc si
mov dl,6e
mov [si],dl
inc si
mov dl,64
mov [si],dl
inc si
mov dl,61
mov [si],dl
inc si
mov dl,24
mov [si],dl
mov ah,00
int 16
cmp al,00
jne *
mov ah,09
mov dx,0200
int 21
* mov ah,4c
int 2
```

