

Unidad IV: Programación de dispositivos

4.1 El buffer de video en modo texto

Se llama modo texto a cualquier modo de vídeo de un ordenador en el que el contenido de la pantalla se representa internamente en términos de caracteres textuales en lugar de píxeles individuales. Típicamente, la pantalla consistirá en una rejilla uniforme de «celdas de caracteres», cada una de las cuales contendrá un carácter del juego de caracteres.

El uso del modo texto se hizo popular a principios de los años 1970, cuando los terminales de texto orientados a vídeo empezaron a reemplazar a los teletipos en el uso interactivo de las computadoras.

4.2 Acceso a discos en lenguaje ensamblador

Vamos a señalar las funciones que debe realizar un computador para ejecutar trabajos de entrada/salida:

- **Direccionamiento o selección del dispositivo** que debe llevar a cabo la operación de E/S.
- **Transferencia de los datos** entre el procesador y el dispositivo (en uno u otro sentido).
- **Sincronización y coordinación** de las operaciones.

Esta última función es necesaria debido a la diferencia de velocidades entre los dispositivos y la CPU y a la independencia que debe existir entre los periféricos y la CPU (por ejemplo, suelen tener relojes diferentes).

Se define una transferencia elemental de información como la transmisión de una sola unidad de información (normalmente un byte) entre el procesador y el periférico o viceversa. Para efectuar una transferencia elemental de información son precisas las siguientes funciones:

- **Establecimiento de una comunicación física** entre el procesador y el periférico para la transmisión de la unidad de información.
- **Control de los periféricos**, en que se incluyen operaciones como prueba y modificación

del estado del periférico. Para realizar estas funciones la CPU gestionará las líneas de control necesarias.

Definiremos una operación de E/S como el conjunto de acciones necesarias para la transferencia de un conjunto de datos (es decir, una transferencia completa de datos). Para la realización de una operación de E/S se deben efectuar las siguientes funciones:

- **Recuento de las unidades de información** transferidas (normalmente bytes) para reconocer el fin de operación.
- **Sincronización de velocidad** entre la CPU y el periférico.
- **Detección de errores** (e incluso corrección) mediante la utilización de los códigos necesarios (bits de paridad, códigos de redundancia cíclica, etc.)
- **Almacenamiento temporal de la información.** Es más eficiente utilizar un buffer temporal específico para las operaciones de E/S que utilizan el área de datos del programa.
- **Conversión de códigos**, conversión serie/paralelo, etc.

4.3 Programación del puerto serial

Comunicación serial en lenguaje ensamblador.

En lenguaje ensamblador, se puede acudir a la interrupción 14H de la ROM-BIOS para configurar, leer, escribir o simplemente para conocer el estado del puerto; cada una de estas cuatro opciones es un servicio de la interrupción, y se seleccionan a través del registro AH. La figura 1 muestra los valores que debe contener el registro AH para invocar cada uno de los servicios. En todos los casos, el registro DX debe contener el número del puerto serie; el primero de ellos, COM1 se especifica como 00h.

Servicio	Descripción
00	Inicializar puerto serie
01	Enviar un dato
02	Recibir un dato

figura 1. Servicios para el puerto serial disponibles a través de la interrupción 14H

Para configurar o inicializar el puerto serie, bastará con utilizar el servicio 00 de la interrupción, colocando en el registro AL los valores equivalentes a los parámetros, como se puede observar en la figura 2. Con éste método es posible obtener frecuencias de transmisión que van desde los 110 hasta los 9600 baudios.

4.4 Programación del puerto paralelo

En lenguaje ensamblador, se puede leer un dato del puerto mediante la instrucción IN ó escribir un dato en el puerto con la instrucción OUT , en ambos casos el registro AL debe participar activamente en la instrucción, bien sea como fuente (en operaciones de escritura) o destino (en operaciones de lectura) del dato, como en los siguientes casos:

out DX, AL ;lleva al puerto DX el contenido del registro AL

in AL,DX ;lleva al registro AL, el contenido del puerto DX

Otra, exigencia, es que el número del puerto sobre el que se va a realizar la transferencia de datos debe estar señalado por el registro DX, a excepción de los casos en los cuales el número del puerto es inferior a 255 (FFh), en cuyo caso la instrucción que lee o escribe puede señalar directamente el puerto.

Como verán la utilización del puerto paralelo en lenguaje ensamblador es muy sencilla.

4.5 Programación híbrida

La compilación híbrida es utilizada en lo que es conocida como programación híbrida (vélgase la redundancia).

La programación híbrida es utilizada en los casos en donde el código en ensamblador dificulta la estructuración del programa. La programación híbrida proporciona un mecanismo por medio del cual podemos aprovechar las ventajas del lenguaje ensamblador y los lenguajes de alto nivel, todo esto con el fin escribir programas más rápidos y eficientes.

Ejemplo: la red es el siguiente donde utilizamos código en ensamblador y código pascal. El siguiente código sirve para limpiar pantalla, está escrito en ensamblador dentro de una función en pascal, lo que es lo mismo a la función `ClrScr` en pascal.