

LABORATORIO DE PROGRAMACIÓN EN LENGUAJE ENSAMBLADOR x86-16bits

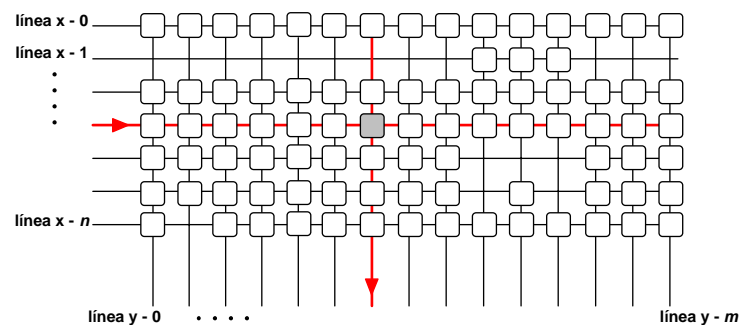
El teclado. Interrupciones BIOS. Servicios de teclado (INT 16h)

Objetivo

El objetivo de esta práctica es conocer el funcionamiento del teclado así como los servicios relativos al teclado que proporciona el BIOS (*Basic Input-Output System*) a través de la interrupción 16h.

Funcionamiento del teclado

Un teclado es una matriz de cables horizontales y verticales cruzados. En cada intersección se coloca un pulsador que permite conectar la línea horizontal con la línea vertical. El procesador de teclado suministra corriente a una línea horizontal y sondea todas las líneas verticales. En caso de que se haya pulsado una tecla en esa línea, habrá una línea vertical que tendrá corriente. Una vez terminado el sondeo de todas las líneas verticales, el procesador suministra corriente a otra línea horizontal y repite el sondeo de las verticales. Y así sucesivamente, con la suficiente frecuencia como para no perder ninguna pulsación.



Cada tecla tiene un código único de 7 bits conocido como *scan code*. Así, con este sistema, podríamos construir teclados que tuvieran hasta 128 teclas. Las teclas de idéntica función (por ejemplo, *Ctrl*, *Shift*, etc.) tienen códigos distintos dependiendo de su posición.

Cuando se pulsa una tecla, el procesador de teclado determina el *scan code* de la misma y lo envía en un byte (MSB a 0 y código de 7 bits) por un interfaz serie similar al RS-232. Cuando la tecla se libera, el procesador de teclado le suma 80h al *scan code* y envía el byte (MSB a 1 y código de 7 bits) por el interfaz serie. De este modo, cada pulsación de tecla provoca el envío de 2 bytes de información por el canal serie: uno para anunciar la pulsación y otro para anunciar la liberación.

Casi todas las teclas tienen la propiedad de que, si se las pulsa de manera continua durante un periodo de tiempo, envían su *scan code* repetido con una frecuencia fija. El retardo antes de entrar en repetición (*delay rate*) suele ser de 0,5s y la frecuencia de repetición (*repeat rate*) típica es de 10 caracteres por segundo. Los primeros teclados sólo enviaban datos y no admitían recepción de comandos, sin embargo, hoy en día, la frecuencia de repetición y otros parámetros son configurables.

Como vemos, el trabajo del teclado –a excepción de recibir comandos de configuración– es tan simple como el envío por un canal serie de códigos de tamaño byte. Al otro lado de la línea serie, hace falta un controlador que recoja el código y lo haga disponible al sistema.

Controlador de teclado

El controlador de teclado es un dispositivo *hardware* compatible con las especificaciones del integrado 8042 de *Intel* (controlador PS/2). Su función es recoger, del flujo serie tipo RS-232, el código (*scan code*) enviado por el teclado, salvarlo en un registro e informar de su disponibilidad al sistema.

El mapa de entrada/salida es el siguiente:

puerto	tipo de acceso	función
60h	lectura	<i>scan code</i>
60h	escritura	programación del controlador o configuración del teclado
64h	lectura	registro de estado
64h	escritura	programación del controlador

El código *scan code* es accesible en lectura por el puerto 60h. La disponibilidad se manifiesta de 2 maneras: o bien en el registro de estado accesible en lectura por el puerto 64h o bien por el disparo de la interrupción *hardware* IRQ 1 (INT 9). Para recibir datos del teclado, existen, por tanto, 2 métodos. El primero y más ineficiente consiste en hacer sondeo sobre el puerto 64h. El bit 0 del registro de estado conmuta a 1 cuando hay un código de tecla disponible. En ese momento, podemos leer el puerto 60h para conocer el *scan code*. El segundo método y el más habitual consiste en ejecutar la rutina de servicio de la IRQ 1. Esta rutina determina el carácter a partir del *scan code* (si es el caso) y lo salva en un *buffer*.

Soporte BIOS del teclado

El BIOS (*Basic Input-Output System*) proporciona soporte lógico para el manejo del teclado. Básicamente, se necesita un *buffer* de memoria que almacene las teclas pulsadas y una copia del estado de algunas teclas especiales.

El *buffer* se organiza como un *buffer* circular con dos punteros, el puntero *head* (última posición escrita) y el puntero *tail* (última posición leída). Por cada tecla, el *buffer* de teclado almacena 2 bytes: el *scan code* y el ASCII. Si la tecla en cuestión no tiene código ASCII, en su lugar se almacena un 0. El BIOS reserva por defecto un *buffer* de teclado para 16 teclas, es decir, 32 bytes. Sin embargo, permite definir *buffers* más grandes si se desea.

Respecto al estado del teclado, es necesario registrar si algunas teclas están pulsadas o no. Por ejemplo, la tecla de bloqueo de mayúsculas, la tecla insertar, etc. De esta manera, por ejemplo, cuando llega un *scan code* de una tecla alfabética podemos determinar su ASCII en función de si la tecla de bloqueo de mayúsculas está pulsada o no. O, si se ha pulsado una combinación de la tecla *Ctrl* con otra, podemos asignar un ASCII adecuado¹.

La siguiente tabla muestra las entradas en memoria más importantes del área de datos del BIOS relativas al teclado.

dirección	función
0040:0017	byte de estado de teclado 1
0040:0018	byte de estado de teclado 2
0040:001A	puntero <i>head</i>
0040:001C	puntero <i>tail</i>
0040:001E	<i>buffer</i> de teclado de 32 bytes
0040:0080	puntero al comienzo del <i>buffer</i> de teclado alternativo
0040:0082	puntero al final del <i>buffer</i> de teclado alternativo

Existen otras posiciones del área de datos del BIOS que almacenan información sobre el tipo de teclado, estado de los LEDs del teclado (si los tiene), configuración del *reset* por *Ctrl-Alt-Del*, etc.

A continuación se muestra la codificación de los bytes de estado:

byte de estado 1	significado	byte de estado 2	significado
x x x x x x x 1	<i>Shift</i> derecha pulsado	x x x x x x x 1	<i>Ctrl</i> izquierda pulsado
x x x x x x 1 x	<i>Shift</i> izquierda pulsado	x x x x x x 1 x	<i>Alt</i> izquierda pulsado
x x x x x 1 x x	<i>Ctrl</i> pulsado	x x x x x 1 x x	<i>Sys req</i> pulsado
x x x x 1 x x x	<i>Alt</i> pulsado	x x x x 1 x x x	<i>Pause</i> activado
x x x 1 x x x x	<i>Scroll lock</i> activado	x x x 1 x x x x	<i>Scroll lock</i> pulsado
x x 1 x x x x x	<i>Num lock</i> activado	x x 1 x x x x x	<i>Num lock</i> pulsado
x 1 x x x x x x	<i>Caps lock</i> activado	x 1 x x x x x x	<i>Caps lock</i> pulsado
1 x x x x x x x	<i>Insert lock</i> activado	1 x x x x x x x	<i>Insert lock</i> pulsado

¹ Algunas combinaciones de teclas tienen un *scan code* distinto en función de la combinación. Es el caso de las teclas de función *Fn* en combinación con las teclas de control (*Ctrl*) o las alternativas (*Alt*). En estos casos, el procesador de teclado envía códigos diferenciados.

La interrupción de teclado IRQ 1 o INT 9

El método habitual para leer el teclado es ejecutar la rutina de atención a la interrupción *hardware* IRQ 1 que se corresponde con el nivel 9 en la tabla de vectores de interrupción (INT 9). Esta interrupción se dispara automáticamente cada vez que el controlador de teclado tiene un dato disponible. Como dato curioso comentaremos que, aunque las interrupciones enmascarables estén deshabilitadas (no se atiende ninguna IRQ), el controlador de teclado es capaz de almacenar 16 teclas y pasárselas al *buffer* de teclado.

La tarea de la rutina de atención a la interrupción de teclado es rellenar el *buffer* de teclado con las teclas pulsadas. Por cada tecla, guarda 2 bytes: el *scan code* y el ASCII si lo tiene (0 si no lo tiene). Adicionalmente, actualiza los bytes de estado del teclado y modifica algunos datos del área de memoria del BIOS si es necesario. Seguidamente se enumeran de manera breve las diferentes acciones de la rutina.

- si se pulsa una tecla ASCII, la rutina copia en el *buffer* de teclado el *scan code* y el ASCII, y actualiza el puntero sumando 2
- si se pulsa *Ctrl*, *Shift* o *Alt*, se actualizan los bytes de estado
- si se pulsa *Ctrl-Alt-Del* se actualiza el *flag* de *reset* del BIOS
- si se pulsa *Pause* se entra en un bucle hasta que no se pulsa una tecla ASCII
- si se pulsa *Print Screen* se invoca la interrupción INT 5
- si se pulsa *Ctrl-Break* se invoca la interrupción INT 1Bh

Dentro de la rutina de servicio a la INT 9 y antes de escribir nada en el *buffer* de teclado, se invocan algunos servicios de la INT 15h con el fin de permitir modificar el significado de la tecla pulsada.

Servicios de la interrupción 16h

Los servicios de la interrupción BIOS INT 16h se encargan de proporcionar los códigos tecleados a las aplicaciones. Digamos que si la tarea de la INT 9 es escribir en el *buffer* de teclado, la INT 16h tiene como tarea leer dicho *buffer*.

Disponemos de un conjunto de servicios muy amplio: lecturas de estado, lecturas de códigos, programación de parámetros del teclado, vaciado del *buffer* de teclado, etc. para una gran variedad de modelos de teclado. En total, decenas de servicios. Los servicios 00h, 01h y 02h realizan la lectura del *buffer* de teclado, la lectura del estado del *buffer* y la lectura del byte de estado 1 respectivamente para teclados antiguos de 82 teclas. Veremos a continuación dichos servicios para teclados expandidos (algo más de 100 teclas).

Servicio 10h – lectura bloqueante de carácter

El servicio 10h de la INT 16h lee los códigos asociados a una tecla del *buffer* de teclado expandido, los escribe en el registro acumulador (AX) e incrementa el puntero del *buffer* de teclado. Si el *buffer* está vacío espera a que se pulse una tecla.

argumentos

ah = 10h

valores devueltos

al = código ASCII

ah = *scan code*

Servicio 11h – lectura no bloqueante de carácter

El servicio 11h de la INT 16h devuelve el estado del *buffer* de teclado expandido, es decir, si está vacío o no. En caso de que no esté vacío, devuelve en AX los códigos asociados sin actualizar el puntero del *buffer* de teclado. Es decir, sucesivas lecturas con este servicio, leen siempre lo mismo.

argumentos

ah = 11h

valores devueltos

zf = 0 si hay carácter y 1 si el *buffer* está vacío

si el *buffer* no está vacío: al = código ASCII

ah = *scan code*

Servicio 12h – lectura del byte de estado 1

El servicio 12h de la INT 16h devuelve el byte de estado 1 del teclado expandido.

argumentos: ah = 12h

valores devueltos: al = byte de estado 1

Prácticas

A) Acceso al teclado por sondeo.

Realice un programa que acceda al controlador de teclado por sondeo y muestre en pantalla el *scan code* en hexadecimal de cada tecla pulsada.

En primer lugar, se deben desactivar las interrupciones *hardware* (instrucción CLI) de manera que la rutina de servicio de teclado no capture la pulsación de las teclas. Antes de finalizar el programa y devolver el control al sistema, se deben habilitar las interrupciones de nuevo (instrucción STI).

Se sondea por programa el puerto 64h continuamente. Si hay pulsación o liberación de tecla se indicará en el bit menos significativo (LSB) del byte de estado. En ese caso, se lee el *scan code* del carácter en el puerto 60h. Cuando MSB = 0 indica pulsación y cuando MSB = 1 indica liberación. Si no se actúa sobre el teclado, se sigue sondeando.

El programa termina cuando se pulsa la tecla del carácter 'S' (*scan code* 1Fh). Utilice la biblioteca de macros para la conversión de entero byte a hexadecimal y para la conversión de dígito hexadecimal a carácter ASCII. También puede utilizar una macro para presentar los *scan code* de las teclas pulsadas en pantalla.

Obsérvese que este método de acceso al teclado, obtiene datos en bruto, es decir, solamente *scan codes*. Para determinar el código ASCII de cada tecla, es necesario un procesamiento adicional que tendrá en cuenta el estado del teclado (mayúsculas, bloqueo de mayúsculas, etc.) así como la codificación regional activa.

B) Acceso al teclado usando los servicios BIOS.

Cuando está habilitada la interrupción de teclado, es esta rutina la que realiza el procesamiento necesario para determinar el código ASCII de cada pulsación. En este caso, se accede al teclado a través del *buffer* de teclado en el que tenemos tanto el *scan code* como el ASCII (si es el caso) de la tecla pulsada.

Como hemos visto, la BIOS nos ofrece una serie de servicios para interactuar con el teclado: sondear el estado del *buffer* de teclado (disponibilidad o no de teclas pulsadas), leer en el *buffer* de teclado los valores tanto de *scan code* como ASCII de cada pulsación y conocer el byte de estado 1.

Realice un programa que sondee la disponibilidad de teclas pulsadas en el *buffer* del teclado. En caso de que haya una tecla disponible, muestre en pantalla su código ASCII, su *scan code* y el byte de estado 1. La salida será similar a esta:

```
Pulse el teclado ('S' termina el programa)
Código ASCII: h   Scan code: 23   Estado: 20
Código ASCII: G   Scan code: 22   Estado: 22
Código ASCII: t   Scan code: 14   Estado: 20
Código ASCII: f   Scan code: 21   Estado: 20
Código ASCII: D   Scan code: 20   Estado: 22
Código ASCII: S   Scan code: 1F   Estado: 22
```

Como se observa en el ejemplo, antes de volcar la información obtenida gracias a los servicios BIOS, se escribe una cadena de caracteres indicando qué es lo que mostramos. El programa finaliza cuando se pulse el carácter 'S'. Utilice la biblioteca de macros.