REPÚBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DE LA DEFENSA
UNIVERSIDAD NACIONAL EXPERIMENTAL POLITÉCNICA
DE LA FUERZA ARMADA
DEP. DE ING. ELECTRÓNICA
LABORATORIO DE ARQUITECTURA DEL COMPUTADOR

Práctica Nº 5 Video y Teclado

Objetivos:

- Diseñar rutinas que permitan realizar operaciones de lecturas a través del teclado.
- Diseñar rutinas que permitan mostrar datos en la pantalla del computador

Marco Teórico:

Pocos programas informáticos adolecen de interacción hombre-maquina, normalmente se requiere que los usuarios del programa introduzcan algún tipo de información u obtengan algún resultado del programa. A diferencia de los lenguajes de programación de alto nivel que disponen de una o de pocas instrucciones que permiten al programador añadir interacción de forma fácil, en Assembler el programador debe diseñar sus rutinas para estas tareas. El BIOS de las PCs dispone de rutinas (interrupciones) que permiten realizar tanto la entrada como la salida de datos ASCII. Para interactuar con otros tipos de datos es necesario implementar rutinas de conversión junto con las rutinas para ASCII.

INTERRUPCIONES:

Una petición de interrupción IRQ ("Interrupt Request") es una señal que se origina en un dispositivo (por ejemplo, un periférico), para indicar al procesador que algo requiere su atención inmediata; se solicita al procesador que suspenda lo que está haciendo para atender la petición. Sin ellas el sistema debería chequear constantemente los dispositivos para comprobar su actividad, pero las interrupciones permiten que los dispositivos puedan permanecer en silencio hasta el momento que requieren atención del procesador.

Las peticiones de interrupción pueden ser generadas no solo por dispositivos hardware, también por los programas, e incluso en circunstancias especiales (errores generalmente) por el propio procesador. Resumimos que existen tres posibles orígenes de estas peticiones: Hardware, Software, y del procesador.

Funcionamiento:

Cuando un dispositivo reclama atención del procesador es para que este haga algo. Este "algo" es lo que se conoce como servicio; controlador o gestor de la interrupción, ISR ("Interrupt service routine"). En cualquier caso se trata siempre de ejecutar un programa situado en algún lugar de la memoria RAM o en la ROM-BIOS. Ocurre que las direcciones de inicio de estos programas, que se conocen como vectores de interrupción, se copian en una tabla de 1024 Bytes que se carga al principio de la memoria de usuario (direcciones 0000h a 0400h) durante el proceso de inicio del sistema, razón por la cual estas rutinas se conocen también como servicios del BIOS. La citada tabla se denomina tabla de vectores de interrupción IDT ("Interrupt Description Table") y en sus 1024 bytes pueden almacenarse 256 vectores de 4 bytes. Es decir, los vectores de interrupción son punteros de 32 bits, numerados de 0 a 255, que señalan las direcciones donde comienza la rutina que atiende la interrupción. El diseño del PC solo permite 16 interrupciones distintas, por lo que puede parecer extraño que se hayan previsto 256 vectores para atenderlas. La razón es que además de los servicios del BIOS propiamente dichos, se cargan las direcciones de inicio de otras rutinas del Sistema Operativo, los denominados servicios del Sistema. Incluso es posible cargar direcciones para rutinas específicas de usuario.

Al diseñar el 8088, Intel estableció un reparto de estos vectores, reservando los 5 primeros para uso interno del procesador (precisamente para atender las excepciones). A continuación estableció otros 27 de uso reservado, aunque no desveló ningún uso específico para algunos de ellos. A partir de aquí, los vectores 32 a 255 estaban disponibles. El esquema resultante se muestra en la tabla adjunta.

Vector		
Dec.	Hex	**Uso**
0	**0**	Error: División por cero
1	**1**	Excepciones para depuración (ejecución paso a paso)
2	**2**	Interrupción no enmascarable
3	**3**	Punto de ruptura interrupción (Instrucción INT)
4	**4**	Desbordamiento ("Overflow"). Utilizado cuando un cálculo aritmético se desborda. Instrucción INTO
5	**5**	(reservado)
6	**6**	Código de instrucción no válido
7	**7**	Coprocesador no disponible
8	**8**	Fallo doble
9	**9**	(reservado -Rutina de atención del Teclado-)
10	**A**	TSS no válido
11	**B**	Segmento no disponible
12	**C**	Excepción de pila
13	**D**	Protección general
14	**E**	Fallo de página
15	**F**	(reservado)
16	**1A**	Error de coprocesador
17-31	**1B-1F**	(reservado)
32-255	**20-FF**	Disponibles para interrupciones enmascarables

Sin embargo, aunque teóricamente las interrupciones 0 a 31 estaban restringidas, IBM y Microsoft utilizaron algunas de ellas sin respetar las indicaciones de Intel. En concreto, IBM y Microsoft utilizaron algunas para los servicios BIOS. Es significativo que, a pesar de haber sufrido ampliaciones, la especificación inicial se mantiene para las 31 interrupciones iniciales. Lo que hace posible que pueda cargarse un Sistema PC-DOS 1.0 en una máquina Pentium IV.

El "modus operandi" es como sigue: Cuando se recibe la petición de interrupción, el procesador termina la instrucción que está ejecutando; guarda el contenido de los registros; deshabilita el sistema de interrupciones; ejecuta el "servicio", y vuelve a su punto de ejecución. El servicio suele terminar con una instrucción IRET ("Interrupt Return") que restituye el contenido de los registros y vuelve a habilitar el sistema de interrupciones.

Servicios del Sistema:

La tabla de vectores de interrupción IDT ("Interrupt Description Table") del PC contiene además de los servicios del BIOS los denominados servicios del Sistema; quedando espacio para la inclusión de servicios especiales del usuario. Como es natural, los vectores de servicios del BIOS señalan a direcciones de memoria situadas en el espacio de la propia BIOS y son idénticos para un modelo determinado de ordenador, con independencia del SO cargado. Por ejemplo: MS-DOS; Windows o Linux. Mientras que los vectores de servicios del Sistema dependerán naturalmente del SO cargado y señalan a zonas de memoria RAM.

Detalle de servicios del MS-DOS:

La tabla de vectores de interrupción del PC tiene posiciones que son estándar, esto significa que algunos números de interrupción corresponden a un mismo servicio en todas las máquinas. En la tabla adjunta se han señalado algunos de estos servicios, indicando el número de interrupción, la dirección del vector y el uso de la ISR correspondiente.

Número	Dirección	Uso
0	0000h	Cuando se realiza una división por cero, el procesador genera una excepción que tiene este número.
1	0004h	.Se utiliza para ejecutar programas paso a paso (depuradores).
2	0008h	Generado cuando se produce una interrupción no enmascarable NMI.
5	0014h	Servicio de impresión del contenido de la pantalla. La invocación de este servicio provoca el mismo resultado que la combinación de teclas SHIFT+CTRL (que utilizan este servicio).
8	0020h	El reloj del sistema genera interrupciones con este número con una frecuencia de 18.21 veces por segundo aproximadamente (se conocen como "ticks"). Este servicio incrementa en una unidad la cuenta del reloj (almacenada en la memoria de datos de la ROM-BIOS). Del valor de este contador, que es puesto a cero cada 24 horas, se basan los servicios de hora del sistema.

Número	Dirección	Uso
16	0040h	Servicios de video. Contiene 16 subservicios tales como ajuste del tamaño y desplazamiento del cursor, escritura y lectura de un carácter o de un píxel, desplazamiento vertical ("Scroll"), ajuste y lectura del modo de video, etc.
17	0044h	Servicio de componentes del equipo. Proporciona una palabra de 16 bits que contiene información básica sobre los componentes instalados en el ordenador.
18	0048h	Este servicio informa del tamaño de la memoria instalada en el sistema.
19	004Ch	La ROM BIOS proporciona seis servicios estándar de disquete: Reinicialización; obtención del estado; lectura; escritura y verificación de sectores, y formateo de pistas.
20	0050h	En esta interrupción se agrupan diversos subservicios de comunicaciones para puertos serie proporcionados por la BIOS: Inicializar los parámetros de inicio del puerto; enviar un carácter; recibir un carácter; obtener el estado del puerto.
21	0054h	Incluye los servicios de cassette. En realidad es una reliquia prehistórica, ya que el PC tenía posibilidad de una cassette opcional.
22	0058h	Incluye los servicios de teclado que incluye tres subservicios: Lectura de un carácter del buffer del teclado; Informar si hay algún carácter en el buffer; selección mayúsculas/minúsculas (estado del "Shift").
23	005Ch	Incluye tres subservicios para impresora: Enviar un carácter al puerto paralelo; Inicializar el puerto; obtener un informe esquemático del estado de la impresora.
24	0060h	Esta es otra reliquia prehistórica. Se trata del cargador del intérprete BASIC; este lenguaje que estaba presente en la ROM de los primeros PC's.
25	0064h	Este servicio es la rutina de puesta en marcha ("bootstrap") del equipo. El resultado de su activación es equivalente a su puesta en marcha. Se denomina reinicio en caliente, y es parecido al efecto que se consigue por el teclado con la combinación CTRL+ALT+DEL.
26	0068h	Este servicio suministra la hora del sistema mediante la inspección del contador de ticks de reloj pasados desde media noche. Incluye dos subservicios, que permiten respectivamente leer y escribir el valor del contador de ticks. **Nota**: La librería estándar C++ dispone de una función **clock**() que devuelve el valor del tiempo transcurrido desde el inicio del programa. Esta función se basa en la inspección del contador de ticks.

Aunque la tabla de vectores de interrupción acepta 256 entradas, esto no es suficiente para todas la tareas que hay que realizar, es por ello que algunas interrupciones realizan distintas tareas dependiendo de un valor de entrada que se conoce como Función y que se coloca en un registro antes de invocar a la interrupción.

CONVERSION DE TIPOS DE DATOS:

Cuando se trabaja con datos de distintos tipos es necesario establecer una equivalencia porque de lo contrario no se podrán operar entre ello, por ejemplo: si queremos realizar la siguiente operación matemática: 0100d + 0FFFh debemos o buscar el equivalente en Hexadecimal del 0100d o debemos buscar el equivalente en BCD del 0FFFh, ya que los micros interpretan solo datos binarios y los datos Hexa son directamente equivalente a los binarios, es conveniente llevar los datos a Hexa. Por tanto la operación equivalente a resolver es: 064h+ 0FFFh = 01063h si dicho dato debe ser entregado a un operador es conveniente regresarlo a su valor BCD: 04195d.

Un ejemplo de conversión de tipos de datos es:

El carácter “1” en ASCII es el Numero 49 de BCD o 031 en Hexa por tanto si queremos convertir el “1” ASCII en Hexa debemos restarle 30h,

“1” = 031h - 30h = 1h.

Es responsabilidad del programador hacer que los datos recibidos y entregados estén plasmados de forma clara, por ejemplo si se esta esperando que el operador presione “1” o “2” para seleccionar una tarea debe informársele mediante un mensaje. Un mensaje del tipo “Introduzca su selección:” seria inadecuado, ya que no se le indica al operador cual serian sus posibles opciones y que tarea se realiza con cual selección, un mensaje adecuado seria: “Presione 1 para hacer la tarea A o Presione 2 para realizar la tarea B. Los datos a introducir deben adecuarse al

nivel intelectual del operador por ello se recomienda que los datos a introducir sean en BCD y usando el sistema métrico adecuado.

Pre-Laboratorio:

Investigue las siguientes funciones/interrupciones

- Función 0 Interrupción 16h
- Función 1 Interrupción 16h
- Función 2 Interrupción 16h
- Función 1 Interrupción 21h
- Función 2 Interrupción 21h
- Función 9 Interrupción 21h

Al programa al final de este documento se le han incorporado varios errores de sintaxis y/o ejecución. Basado en esta información se pide lo siguiente:

- Identifique y corrija los errores hasta que consiga una versión funcional del programa
- Describa el objeto del programa

Desarrollo de la práctica:

- Elabore un programa que reciba por teclado un número del 0 al 15 y muestre por pantalla su equivalente en hexadecimal
- Transcriba, compile y ejecute el programa

```
MODEL SMALL
.DATA

N1          DB ?
N2          DB ?
OP          DB ?
RS          DB ?

.CODE
INICIO:     mov dx,@data
            mov ds,bx

@1:         mov ah,00h
            int 16h

            cmp al,'0'
            jl @1
            cmp al,'4'
            jg @1

            mov ah,02h
            mov dl,al
            int 21h
            sub al,'0'
            mov N1,al

@2:         mov ah,00h
            int 16h

            cmp al,'+'
            je @3
            cmp al,'-'
            jne @2

@3:         mov OP,al
            mov ah,02h
            mov dl,al
            int 21h

@4:         mov ah,00h
            int 16h

            cmp al,'0'
            jl @4
            cmp al,'4'
            jg @4

            mov ah,02h
            mov dl,al
            int 21h
            sub al,'0'
```

```
                mov N2,al

                mov ah,02h
                mov dl,'='
                int 21h

                mov dl,_N1
                mov al,OP

                cmp al,'+'
                jne @5
                add dl,N2
                jmp @6

@5:             sub dl,N2
                ja @6

                neg dl
                mov RS,dl
                mov ah,02h
                mov dl,'-'
                int 21h
                mov dl,R_S

@6:             add dl,'0'
                mov ah,02h
                int 21h

                mov ah,02h
                mov dl,13
                int 21h
                mov dl,10
                int 21h


                mov ah,4C00h
                int 21h
END             _INICIO
```