

Arquitectura de Computadores I

Laboratorio 1 (solución): interrupciones

Queremos programar un juego de dardos basado en la arquitectura del PC. Cada vez que un dardo impacte en la diana, el dispositivo interrumpe por la línea IRQ3 e indica en su registro de datos (dirección 0x281) la puntuación obtenida. Vamos a suponer que todos los dardos impactan en la diana.

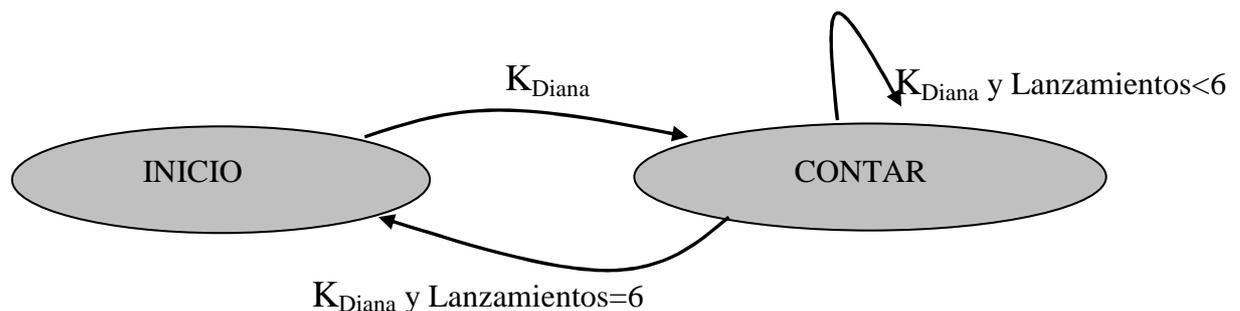
El funcionamiento del juego es muy sencillo: queremos saber qué jugador obtiene la mayor puntuación en el lanzamiento de 6 dardos, teniendo en cuenta además el tiempo invertido. El tiempo empezará a contarse desde el impacto del primer dardo. Al iniciarse el programa sacaremos un mensaje indicando al jugador el comienzo del juego, a fin de que empiece a lanzar los dardos. El programa finalizará indicando mediante otro mensaje la puntuación obtenida y el tiempo empleado (en segundos). Para sacar los mensajes, suponed ya implementadas las funciones `EscribeTexto(texto)` y `EscribeNum(num)`.

Teniendo en cuenta el funcionamiento descrito escribe en el lenguaje C las siguientes funciones:

- Programa principal. Indica cuales son las variables globales necesarias para el funcionamiento del programa.
- Todas las rutinas de atención que consideres necesarias.
- La rutina que modifica el vector de interrupciones para atender las peticiones de un periférico.

Solución

Para resolver el ejercicio utilizaremos el autómata de la figura. El sistema se encontrará en el estado INICIO hasta que se lance el primer dardo y éste impacte en la diana. A partir de ese momento, en el estado CONTAR, se contarán los segundos que pasan y el número de lanzamientos del dardo. Una vez que el número de lanzamientos ha llegado a 6 se vuelve al estado INICIO ya que la partida ha terminado.



- Programa principal. Indica cuales son las variables globales necesarias para el funcionamiento del programa.

En el programa principal se deben definir e inicializar las variables globales. Además se debe cambiar el vector de interrupciones para que se ejecuten las rutinas de atención a los dispositivos que nosotros hemos programado, que en este caso son la rutina de atención al reloj (se debe modificar la posición 0x1C del vector de interrupciones guardando la dirección de la rutina que hemos escrito) y la rutina de atención a la diana (se debe modificar la entrada 0x0B del vector de interrupciones, la correspondiente a la línea de interrupciones 3). Además, como el programa no tiene fin, en el programa principal tenemos un bucle que no termina nunca. En este bucle lo único que se necesita hacer es mostrar por pantalla el mensaje

“comenzar partida“ si estamos en el estado INICIO (la partida no ha comenzado todavía). Finalmente, en el programa principal, en general, se debe recuperar el estado del vector de interrupciones para lo que se utiliza la función `RecuperaVI`, aunque en este caso, esta función no se ejecutará nunca debido al bucle infinito que tiene por delante.

La variable que guarda el estado del autómata (EA) tiene que ser una variable global, ya que se accede tanto desde el programa principal como desde las dos rutinas de atención que vamos a escribir en este ejercicio. La variable que se utiliza para guardar los segundos transcurridos también va a ser una variable global: se incrementa en la rutina de atención al reloj y su valor se saca por pantalla en la rutina de atención a la diana cuando llegue a valer 6. Las variables utilizadas para guardar el número de lanzamientos y la suma de las puntuaciones pueden ser variables globales o se pueden definir como variables `static` dentro de la rutina de atención a la diana.

```
int EA;
int Suma, Lanzamientos, Segundos;
main()
{
EA = INICIO;
Suma=0;
Lanzamientos=0;
Segundos=0;
unsigned short IPoldDiana, CSoldDiana, IPoldReloj, CSoldReloj;

CambiaVI(0x1C, IP(RutAtencionReloj), CS(RutAtencionReloj),
        &IPoldReloj, &CSoldReloj);
CambiaVI(0x0B, IP(RutAtencionDiana), CS(RutAtencionDiana),
        &IPoldDiana, &CSoldDiana);

while(1)
    if (EA == INICIO) EscribeTexto( "comenzar partida");

RecuperaVI(0x1C, IPoldReloj, CSoldReloj);
RecuperaVI(0x0B, IPoldDiana, CSoldDiana);
}
```

b) Todas las rutinas de atención que consideres necesarias.

Son dos las rutinas de atención a dispositivos que tenemos que escribir: la rutina de atención al reloj y la rutina de atención a la diana.

En la **Rutina de atención del reloj** tenemos que contar el tiempo que llevamos jugando en la partida, es decir tenemos que contar el tiempo que llevamos en el estado CONTAR. Para ello, mediante la variable `Tick` contamos cuando ha pasado un segundo (habrá pasado un segundo cuando hayan ocurrido 18 interrupciones de reloj) y la utilizamos para incrementar la variable `Segundos` cada vez que pasa un segundo y estamos en el estado CONTAR.

```

void interrupt RutAtencionReloj()
{
    static int Tick=0;
    Tick++;
    if (Tick==18)
    {
        Tick=0;
        if (AE==CONTAR)
            Segundos++;
    }
    //IRET;
}

```

Cuando se ejecuta la **rutina de atención a la diana** quiere decir que un dardo ha dado en la diana y queremos saber cual es la puntuación obtenida. Para ello leemos el registro de datos del controlador de la diana y acumulamos la puntuación de este lanzamiento en la variable Suma que al final tendrá la puntuación acumulada de 6 lanzamientos. También se incrementa la variable Lanzamientos. Además, si cuando se produce la interrupción el estado del autómatas es INICIO quiere decir que estamos ante el primer lanzamiento de una partida por lo que se debe inicializar la variable Segundos y cambiar el estado del autómatas a CONTAR. Si por el contrario estamos en el estado CONTAR cuando se produce la interrupción, comprobaremos si es el último lanzamiento (Lanzamientos==6) para sacar por pantalla la información correspondiente y volver a inicializar las variables globales.

```

void interrupt RutAtencionDiana()
{
    Suma= Suma + InPort(0x281);
    Lanzamientos ++;
    if (EA==INICIO)
    {
        EA=CONTAR;
        Segundos = 0;
    }
    else
    {
        if (Lanzamientos==6)
        {
            EscribeTexto("Puntuación");
            EscribeNum (Suma);
            EscribeTexto("Tiempo");
            EscribeNum(Segundos);
            Suma=0;
            Lanzamientos=0;
            EA=INICIO;
        }
    }
    Eoi();
    //IRET;
}

```

c) La rutina que modifica el vector de interrupciones para atender las peticiones de un periférico.

La rutina que modifica el vector de interrupciones debe escribir en la posición del vector de interrupciones indicada como parámetro (entrada) los valores IPNuevo y CSNuevo, guardando previamente los valores de esa posición en las variables IPAnt y CSAnt. Como los valores a leer y escribir son de 16 bits se utilizan las funciones LeerPalFis y EscribePalFis para leer y escribir del vector de interrupciones. El primer parámetro de estas funciones hace referencia a la dirección base del vector de interrupciones en memoria, el segmento 0 concretamente. A la hora de calcular el desplazamiento de la posición o entrada que se quiere acceder hay que tener en cuenta que cada entrada del vector de interrupciones ocupa 4 bytes en memoria (2 bytes el IP y otros dos el CS).

```
void CambiaVI (int entrada, unsigned short IPNuevo,
               unsigned short CSNuevo, unsigned short * IPAnt,
               unsigned short * CSAnt)
{
  DisableInts(); //Deshabilitar interrupciones (IF=0)
  *IPAnt = LeePalFis(0, entrada*4); //Se guarda el IP anterior
  *CSAnt = LeePalFisFis(0, (entrada*4)+2); //Se guarda el CS anterior
  EscribePalFis(0, entrada*4, IPNuevo); //Se escribe el nuevo IP
  EscribePalFis(0, (entrada*4)+2, CSNuevo); //Se escribe el nuevo CS
  EnableInts(); // Habilitar interrupciones (IF=1)
}
```