

Universidad Simón Bolívar.  
Departamento de Computación  
CI-3815 Organización del Computador.  
Enero-Marzo 2015

## **Proyecto # 3 (10 %)**

### **Interrupciones**

## **1 Objetivo**

El objetivo de este proyecto es que el estudiante se familiarice con los principios básicos de interrupciones y entradas y salidas asignadas a memoria en lenguaje ensamblador MIPS

Se desea simular un Reloj usando interrupciones producidas por teclado. Específicamente, usaremos entradas de ciertos caracteres desde el teclado para generar interrupciones que permitan incrementar el Reloj, reinicializar el Reloj o salirse del programa.

## **2 Especificación**

Para la implementación de este proyecto será necesario escribir un manejador de interrupciones que guíe las interrupciones generadas. Puede utilizar el archivo suministrado (exceptions.asm) como esqueleto para escribir su manejador. Su manejador debe:

1. Ser almacenado en un archivo llamado Relojhandler.asm . Este archivo NO debe contener una función main.
2. Ser capaz de manejar las interrupciones del dispositivo teclado usando la técnica de asignación a memoria (Memory-Mapped I/O) y Traps.
3. Contener un segmento .ktext 0x80000180 como aparece en el exceptions que le suministramos.

Cuando su manejador reciba una interrupción del teclado, éste deberá usar la técnica de Entrada/Salida Asignada a Memoria para leer el valor ASCII del caracter que fue introducido. Usted NO podrá usar un syscall para leer el caracter desde el teclado.

Para que la interrupción ocurra y pueda ser atendida se debe tener habilitadas la interrupciones y habilitado el dispositivo. Para ello use la interrupción por software(trap) syscall con código 100. También uselo para inicializar el Reloj en 0 minutos, 0 segundos.

Cada vez que se reciba una entrada desde el teclado, el reloj se verá afectado. Reloj debe ser una variable global.

En particular usaremos tres caracteres: t, q y r. El resto de los caracteres serán ignorados por el manejador. En respuesta a estos caracteres, el manejador deberá llevar a cabo las siguientes acciones:

1. **t (Tick)** Cuando se reciba el caracter t, el manejador deberá incrementar el Reloj en un segundo e imprimir la hora actual (XX:XX)
2. **r (Reset)** El manejador debe reinicializar el Reloj a 0 minutos 0 segundos e imprimir la hora (00:00)
3. **q (Quit)** El manejador deberá salirse del programa usando la llamada al sistema con código 10

### 3 Como funciona

Considere el programa test.asm. Si corre este programa con el manejador que trae el simulador por defecto, usted obtendrá la siguiente salida:

```
Comenzo a ejecutar el programa de prueba
El resultado de la suma es 210
El resultado de la suma es 210
El resultado de la suma es 210
El resultado de la suma es 210
```

```
.... se detiene la ejecucion con STOP
```

El programa test.asm suma los elementos de un arreglo de 20 elementos e imprime el resultado. Después de la impresión vuelve a sumar dichos elementos nuevamente y así continua indefinidamente. Para que el cálculo se demore un poco más se le agregó un lazo para retardar la ejecución. Cuando ocurre una interrupción, el CPU atiende el requerimiento ejecutando el manejador de interrupciones para luego regresar al programa interrumpido. El programa test.asm les será suministrado

Si corremos este mismo programa test.asm usando el nuevo manejador de interrupciones (Relojhandler.asm), la salida a obtener debe ser de la siguiente forma:

```
Comenzo a ejecutar el programa de prueba
El resultado de la suma es 210
El resultado de la suma es 210
Time: 00:01                      ---> Se presiono la tecla t
El resultado de la suma es 210
Time: 00:02                      ---> Se presiono la tecla t
Time: 00:03                      ---> Se presiono la tecla t
El resultado de la suma es 210
Time: 00:00                      ---> Se presiono la tecla r
```

La impresión de la hora actual se debe realizar tanto en la consola de MARS como en el display del MMIO.

Para usar su nuevo manejador de interrupciones debe especificar en la opción Settings el archivo de trap handler que desee utilizar.

## **Entrega**

1. Fecha: Viernes 06 de Marzo de 2015. Deberá subir el código de su proyecto a Moodle antes del viernes 06
2. El código impreso del manejador desarrollado, completamente documentado con explicaciones claras de lo que hace el manejador