

ESTRUCTURA DE COMPUTADORES

NOMBRE Y APELLIDOS	GRUPO
David Arnal García	2D2

Práctica 13: Sincronización por interrupciones

Paso 1: Habilitación de interrupciones y modo de funcionamiento

► **Cuestión 1.** Código de inicio correcto.

```
la $t0, 0xFFFF0000
li $t1, 2
sb $t1, 0($t0)
li $t0, 0x103
mtc0 $t0, $12
```

► **Cuestión 2.** ¿Por qué se producen los mensajes de error?

Porque no está declarado \$k0 en el manejador y, por tanto, se produce la excepción.

Paso 2: Obtención de la dirección de retorno al programa de usuario

► **Cuestión 3.** Copia aquí la línea de código que escribe en \$k0 la dirección de retorno

```
mfc0 $k0, $14
```

Paso 3: Tratamiento provisional de la interrupción de teclado

► **Cuestión 4.** ¿Por qué se escriben tantos asteriscos al pulsar una tecla?

Porque no se cancela la interrupción al pulsar la tecla.

Paso 4: Cancelación de la interrupción

► **Cuestión 5.** Líneas de código que cancelan la interrupción.

```
la $t0, 0xFFFF0000  
lw $a0, 4($t0)
```

► **Cuestión 6.** ¿Por qué acaba el programa de usuario antes de lo esperado?

Porque se está modificando el registro \$t0 sin haber guardado su contenido y, al volver al programa, hay un número negativo, lo que hace que el bucle acabe.

Paso 5: Gestión del contexto

► **Cuestión 7.** Modificaciones:

- En el segmento de datos del manejador:

```
contexto: .word 0x80000400
```

- En el código de inicio del sistema:

```
la $k1, contexto
```

- Al principio del código del manejador:

```
.set noat  
sw $at, 0($k1)  
.set at  
sw $t0, 4($k1)  
sw $a0, 8($k1)  
sw $v0, 12($k1)
```

- Al final del código del manejador:

```
.set noat  
lw $at, 0($k1)  
.set at  
lw $t0, 4($k1)  
lw $a0, 8($k1)  
lw $v0, 12($k1)
```

Paso 6: Habilitación de las interrupciones del reloj

► Cuestión 8. Archivo *teclado-y-reloj.handler*.

- En el código de inicio: instrucciones que habilitan la interrupción del reloj.

```
la $t2, 0xFFFF0010 #DB del reloj
li $t1, 1
sb $t1, 0($t2)
```

- En el código de inicio: desenmascara la línea de interrupción *int2**

```
li $t0, 0x403
```

► Cuestión 9. Explica por qué la consola mostrará asteriscos sin parar.

Porque no se cancela la interrupción, como pasaba en el anterior caso.

Paso 7: Análisis de la causa de excepción

► Cuestión 10. Archivo *teclado-y-reloj.handler*.

- Código a partir de *retexc*:

```
.set noat
lw $at, 0($k1)
.set at
lw $t0, 4($k1)
lw $a0, 8($k1)
lw $v0, 12($k1)
mfc0 $k0, $14
rfe
jr $k0
```

- Tratamiento de la interrupción del teclado.

```
int0:
li $v0, 11
li $a0, '*'
syscall
la $t0, 0xFFFF0000
lw $a0, 4($t0)
j retexc
```

- Tratamiento del reloj.

```
int2:  
li $t0, 0xFFFF0010  
li $a0, 1  
sb $a0 0($t0)  
j retexc
```

- Instrucciones que leen y aíslan el código de causa de excepción.

```
mfc0 $k0, $13  
andi $t0, $k0, 0x003C  
bnez $t0, retexc
```

- Instrucciones que analizan los bits *IP0* e *IP2* de la palabra de estado

```
andi $t0, $k0, 0x400  
bnez $t0, int0  
andi $t0, $k0, 0x1000  
bnez $t0, int2  
beqz $t0, retexc
```