

ESTRUCTURA DE COMPUTADORES

Práctica 14: Llamadas al sistema operativo (I)

NOMBRE Y APELLIDOS	GRUPO
Fabián, Scherle Carboneres	2C2

Cuestión 1. Teniendo *Usuario0.s* en ejecución, detenga el simulador. (Antes de responder sí en el cuadro “Execution paused by the user at PC = ...”, asegúrese que el *PC* apunta a una instrucción del programa de usuario, cuyas direcciones son del tipo “0x0040nnnn”. Si apunta a direcciones del tipo “0x8000nnnn” pulse que no y vuelva a intentarlo).

► ¿Qué vale el registro Status (registro \$12) que aparece en la ventana superior del simulador?

Vale 0x00000003.

► ¿En qué modo está funcionando el procesador? ¿Están habilitadas las interrupciones?

Está funcionando en modo usuario, y están habilitadas las interrupciones, bit KU actual a 1 y bit IE actual a 1.

► ¿Qué valen los bits de máscara de interrupción?

Valen 0 todos los bits de máscara.

► Indique con qué instrucciones del código de inicio de se *MiMoSv0.handler* se inhiben las interrupciones en los periféricos teclado, reloj y consola.

Se inhiben con las siguientes intrucciones:

```
teclado: li $t0, 0xffff0000
          sb $zero, 0($t0)
reloj:    li $t0, 0xffff0010
          sb $zero, 0($t0)
consola:  li $t0, 0xffff0008
          sb $zero, 0($t0)
```

Se carga \$zero (0x00000000) en los registros de estado/control de esta manera se queda el bit Interrupt Enable a 0.

► Indique cómo se inicializa el registro de estado del coprocesador 0.

```
mfc0 $t0, $12
ori $t0, $t0, 0x0003
mtc0 $t0, $12
```

Se inicializa con interrupciones habilitadas y en modo usuario.

Cuestión 2. Escriba el código de tratamiento de la interrupción del reloj.

```
int2:

    lw $t0, segundos
    addi $t0,$t0,1
    sw $t0, segundos

    #Cancelo la interrupción
    li $t0, 0xffff0010
    li $t1, 0x1
    sw $t1, 0($t0)

    b retexc      #fin
```

Cuestión 3. Modifique el código de inicio del sistema para que la interrupción del reloj quede habilitada, tanto en la interfaz como en el registro de estado del coprocesador 0.

```
Habilito la interrupción en la interfaz:
li $t0, 0xffff0010
li $t1, 0x1
sb $t1, 0($t0) # habilito interrupcion en el HW del reloj

Habilito la interrupción en el registro de estado del coprocesador 0:
mfc0 $t0, $12
ori $t0, $t0, 0x0403 # Interrupciones del reloj habilitada
mtc0 $t0, $12
```

Cuestión 4. Escriba el código de la función *get_time* a continuación.

```
get_time:

    lw $v0, segundos
    b retexc
```

Cuestión 5. ¿Podría ejecutarse correctamente *Usuario0.s* con el manejador *MiMoSv.1*?

¿Podría ejecutarse correctamente *Usuario1.s* con el manejador *MiMoS v.0*? Razone la respuesta.

Usuario0.s si podría ejecutarse correctamente con el manejador MiMoSv.1, ya que el manejador mantiene las implementaciones de `print_char` y `get_version`.
Usuario1.s no podría ejecutarse adecuadamente con el manejador MiMopsv.0 ya que no tiene habilitadas las interrupciones del reloj, ni implementada `get_time`. Por lo que devolvera siempre el valor que se le pasa por `$v0` que es 91.

Cuestión 6. Escriba el código de la función *wait_time* que se ha implementado.

wait_time:

```
li $t0, ESPERANDO
sw $t0, estado

lw $t0, segundos
add $t0, $t0, $a0
sw $t0, despertador

b retexc
```

Cuestión 7. Escriba el código de la interrupción del reloj, desde la etiqueta *int2*.

```
int2: lw $t0, segundos
      addi $t0,$t0,1
      sw $t0, segundos

      li $t0, 0xffff0010
      li $t1, 0x1
      sw $t1, 0($t0)
      # Cancela la interrupción pero deja habilitadas las interrupciones del reloj

      lw $t0, estado
      li $t1, ESPERANDO
      bne $t0,$t1,finsi
      lw $t0, segundos
      lw $t1, despertador
      bne $t0,$t1,finsi
      lw $t0, estado
      li $t1, LISTO
      sw $t1, estado

      finsi:
      b retexc # fin
```

Cuestión 8. Detenga el programa *usuario2.s* justamente después de haber escrito el tiempo actual (*n segundos*). ¿Qué valen el PC y el registro *Status*?

PC vale 0x00400040
Status vale 0x00000403

¿Qué código se está ejecutando: el manejador, el proceso de usuario o el proceso ocioso?

Se está ejecutando el proceso ocioso.