**ESTRUCTURA DE COMPUTADORES**

**Práctica 14: Llamadas al sistema operativo (I)**

|  |  |
| --- | --- |
| **NOMBRE Y APELLIDOS** | **GRUPO** |

**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 “0x0040*nnnn*”. Si apunta a direcciones del tipo “0x8000*nnnn*” pulse que no y vuelva a intentarlo).

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

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

►¿Qué valen los bits de máscara de interrupción?

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

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

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

*int2:*

**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.

**Cuestión 4.** Escriba el código de la función *get\_time* a continuación.

*get\_time:*

**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 resupuesta.

**Cuestión 6.** Escriba el código de la función *wait\_time* que se ha implementado.

*wait\_time:*

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

int2:

**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**?

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