COMPUTER ORGANISATION

Lab Session 14: System Calls (II)

NAME AND SURNAME

Question 1. Stop the simulation while running *Usuario0.s*. (Before clicking on "Si" button look at "Execution paused by the user at PC = ...", and check if *PC* points to a user program instruction, that means an address of type "0x0040*nnnn*". If it points to an address of type "0x8000*nnnn*" press "No" and try again).

►What is the value of coprocessor `Status` (register $12) that appear on the top simulator panel?

►In what mode is the processor running? Are the interrupts enabled?

►What is the value of the interrupt mask bits?

► Indicate with what instructions on the *MiMoSv0.handler* starting code are the interrupts disabled for the keyboard, clock and console.

► Indicate how the coprocessor state register is initialized.

From now, several *MiMoS* versions will be succesively developed that will include new functions. **A new version will be created always from the previous one, create a copy of the handler code file renamed with the new version number**.

Question 2. Write the code to handle the clock interrupt.

int2:

Question 3. Modify the starting code in such a way that the clock interrupt become enabled, both on the clock interface and on the coprocessor state register.

Question 4. Write the code of function *get_time*.

get_time:

Question 5. *User0.s* can be executed correctly with handler *MiMoSv.1*? *User1.s* can be executed correctly with handler *MiMoS v.0*? Explain your answer.

Question 6. Write the implemented code for *wait_time*.

wait_time:

Question 7. Write the clock interrupt handling code added from label int2.

int2:

Question 8. Stop *User2.s* just after writing the actual time, what is the PC and Status register contents?

What code was running while you stop execution the handler, the main process or the void process?