

SIMULADOR DE UN SISTEMA INFORMÁTICO PRIMITIVO

TAREAS - V0

Tareas iniciales

Inicia una sesión de terminal (putty o MobaXterm) en la máquina `ritchie`. A continuación, copia el directorio `/var/asignaturas/ssoo/2019-2020/V0` en tu directorio personal; o descarga el fichero comprimido del Campus Virtual si no vas a usar `ritchie`.

El trabajo a realizar en los ejercicios siguientes se desarrollará sobre la copia indicada de los ficheros contenidos en el directorio `v0`, dentro de tu directorio personal.

Se pueden utilizar máquinas virtuales Linux o “*Ubuntu en Windows*” si está configurado, en vez de `ritchie`.

Ejercicios

1. Modifica la llamada a la función `ComputerSystem_DebugMessage()` ubicada en la función `Processor_DecompileAndExecuteInstruction`, y lo que sea necesario en el fichero “`messages.txt`”, para que muestre también el contenido del registro PSW, de tal forma que el mensaje (número 3) tenga el aspecto siguiente (el valor de la PSW está en hexadecimal, y cada subrayado “`_`” debe sustituirse por un único espacio en blanco):

```
(PC:_233,_Accumulator:_1_[00000001],_PSW:_00000000)
```

2. Modifica la función `ComputerSystem_PowerOff()` para que muestre el mensaje número 99, usando sección `SHUTDOWN`, con el aspecto siguiente (incluido el color rojo, y sustituyendo cada subrayado “`_`” por un único espacio):

```
END_of_the_simulation
```

3. Modifica la función `Processor_DecompileAndExecuteInstruction()` para añadir una nueva instrucción `MEMADD` al juego de instrucciones del procesador, que tendrá la siguiente sintaxis y semántica:

```
MEMADD operand1 memAddress
```

La ejecución de la instrucción realizará la suma del valor del `operand1` y el contenido de la celda de memoria indicada por `memAddress` y dejará el resultado en el registro acumulador.

Para crear el enumerado correspondiente se añadiría la línea siguiente al final del fichero “`Instructions.def`”

```
INST(MEMADD) // decimal 10 = Hex (0x0a)
```

4. Escribe un programa de usuario almacenado en un fichero `prog-V0-E4` que guarde, sucesivamente, los valores enteros **pares** comprendidos entre el 19 y el 41 en la celda de memoria de la posición 99. Úsense, para ello, instrucciones de salto y la instrucción `MEMADD` anterior.
5. Escribe un programa de usuario almacenado en un fichero `prog-V0-E5` que guarde sucesivamente, los valores de las potencias de 2 de forma ascendente, desde 1 (2^0) hasta 1024 (2^{10}), en la celda de memoria de la posición 64.
6. OPCIONAL si se han terminado correctamente TODOS los anteriores.

Modifica `ComputerSystem.c` para que el simulador ejecute el programa indicado en un fichero cuyo nombre esté como primera línea, en el fichero de nombre: `"nameOfTheProgramToBeExecuted"`.

Si no existiese el fichero llamado `"nameOfTheProgramToBeExecuted"`, o no existiese el fichero cuyo nombre está en la primera línea del fichero, se intentaría ejecutar el programa del fichero original del simulador: `programToBeExecuted`

Por ejemplo, si en el fichero `nameOfTheProgramToBeExecuted` la primera línea contiene:

```
prog-V0-E4
```

Ejecutaría el programa del ejercicio 4.