

**PRINCIPIOS DE COMPUTADORES**  
**Sesión Práctica 1. Utilización del emulador QTSpim y representación de la información**

## **Introducción.**

Esta práctica tiene dos objetivos. Por un lado, debes familiarizarte con un programa denominado QTSpim, que es un emulador de un procesador MIPS. Por otro lado, vas a practicar conceptos relacionados con la representación de la información.

### **QTSpim**

QTSpim es un emulador de un procesador. Esto significa, que el programa nos permitirá simular el funcionamiento de una máquina diferente a la “máquina física” sobre la que ejecutas el programa. Así, aunque QTSpim se ejecute sobre una máquina con un procesador de alguna familia de los fabricantes habituales (Intel, AMD, ...), podrás realizar y visualizar las operaciones de una máquina virtual cuyo procesador sigue fielmente la arquitectura MIPS de 32 bits. Entre otros aspectos podremos:

- Configurar algunas características del procesador.
- Visualizar y modificar los registros del procesador, incluso durante la ejecución de un programa.
- Cargar y ejecutar un programa escrito con el juego de instrucciones de ensamblador MIPS.
- Visualizar el contenido de la memoria de la máquina virtual.

### **Operación básica de QTSpim**

No olvides nunca que en QTSpim hay procedimientos que pueden ser diferentes al de una máquina real. Por ejemplo, como se ha visto en la asignatura, en una máquina real, el proceso de llevar un programa en ensamblador a la memoria del ordenador de forma que esté listo para ser ejecutado involucra varias fases:

1. obtención del código objeto por parte del compilador
2. montaje (linker)
3. carga en memoria y preparación para la ejecución (todo un proceso de carga del programa y los datos del programa en la memoria bajo la responsabilidad del sistema operativo).

En cambio, QTSpim realiza todos estos pasos de una vez. Tras escribir nuestro programa, lo cargamos en QTSpim y si el programa tiene la sintaxis correcta (el emulador puede traducirlo en instrucciones de la máquina MIPS), será ubicado en la memoria de la máquina virtual y estará listo para ser ejecutado.

Puesto que la máquina virtual que utiliza QTSpim es muy sencilla y está orientada justamente a la prueba de programas escritos para la arquitectura MIPS, no dispone de un sistema operativo con las funcionalidades a las que estamos acostumbrados. Cuando el programa se carga en QTSpim, podremos ejecutarlo, pero el programa no se descargará automáticamente de la memoria una vez finalice. Esto quiere decir, que si

hacemos modificaciones en el código fuente y queremos probar las modificaciones, **debemos reiniciar la máquina virtual usando la orden correspondiente en QTSpim.**

## Ejercicios de la práctica 1

a) Identificar en el emulador los siguientes elementos:

- I. El segmento de Datos (Usuario, Kernel y Pila)
- II. El segmento de Instrucciones (Usuario y Kernel)
- III. El contenido de los Registros Enteros
- IV. El contenido de los Registros en Punto Flotante.
- V. La consola del sistema

b) Edita con un editor de textos plano (vi, vim, gedit, kate, visual studio code o el que prefieras) el fichero `practical.s` y sustituye la cadena `"apellido1 apellido2 nombre, NIF, alu123456789@ull.edu.es \n"` con tus apellidos, nombre, NIF (NIE o pasaporte) y dirección de correo y graba el fichero. A continuación, carga el programa en QtSpim y ejecuta el programa de una sola vez. Comprueba que la salida emitida por la consola corresponde con lo que esperabas.

c) Explora el segmento de datos. Comprueba que el segmento de datos está representado en Hexadecimal (en el menú Data Segment marca el checkbox correspondiente a Hex). Con las indicaciones que te dé tu profesor en la práctica responde a las siguientes preguntas:

- I. ¿Qué dirección de memoria (expresa la dirección en hexadecimal) ocupa el primer carácter de tu nombre? (p.ej: en "Martin Galan Carlos, 43777444Z, alu123456789@ull.edu.es\n" el carácter en cuestión es "C")?
- II. ¿Qué carácter es y qué representación tiene en hexadecimal?
- III. Recuerda que estás en hexadecimal. Busca en el segmento de datos de QTspim el número que se encuentra en la dirección etiquetada como `num3`.
- IV. Convierte el número `4.301` a formato IEEE-754 para 32 bits (usa los apuntes del profesor o utiliza una [calculadora online](#)). Busca ahora este número en el segmento de datos.
- V. ¿En qué dirección empieza el número `4.301`? expresa la dirección en hexadecimal.
- VI. Convierte el número `35531561.13` a formato IEEE-754 para 64 bits (usa los apuntes del profesor o utiliza una [calculadora online](#)). Busca ahora este número en el segmento de datos.
- VII. ¿En qué dirección empieza el número `35531561.13`? expresa la dirección en hexadecimal.

**NOTA IMPORTANTE:** si estás utilizando qtspim en linux, es posible que los números en punto flotante estén mal representados. Esto se debe a que el punto o coma decimal varía según los diferentes idiomas (en concreto cambia en español y en inglés).

En la sección de prácticas del campus virtual puedes ver [cómo arrancar el qtspim desde una terminal en Linux](#) y evitar este problema (desde la terminal ejectuar `LC_NUMERIC="en_US.UTF-8" qtspim &` o sencillamente `LANG=C qtspim &`)

d) Reinicia la máquina y vuelve a cargar el programa en el QTSpim. Visualiza el banco de registros enteros y flotantes en hexadecimal (Menú Registers opción Hex). **Recuerda, que las instrucciones de tu programa pueden ser convertidas en una o más instrucciones en QTSpim, por lo que tendrás que buscar la instrucción original de tu programa en los comentarios de la parte derecha.**

- I. Ejecuta paso a paso el programa hasta que hayas encontrado la instrucción `add $t2,$t0,$t1`

Una vez se haya ejecutado analiza el banco de registros enteros, en concreto, el \$t2. ¿Qué valor contiene? ¿sabrías expresarlo en decimal?

- II. Cuando hayas terminado de ejecutar esta instrucción, modifica a mano el valor del registro \$t3 (pulsas con el botón derecho del ratón sobre el registro correspondiente en el banco de registro y seleccionas *“Change Register Contents”*, allí puedes seleccionar el formato y el valor). **Deberás introducir un valor 1200 en formato decimal.**

- III. A continuación sigue ejecutando paso a paso hasta terminar de ejecutar la instrucción `add $t4,$t2,$t3`. ¿Qué valor tiene el registro \$t4 en hexadecimal? ¿y en decimal?

- IV. A continuación establece un punto de ruptura *“breakpoint”* sobre la instrucción `move $a0,$t3` (sobre la instrucción correspondiente, pulsas en el botón derecho del ratón y seleccionas *“Set Breakpoint”*). Después ejecuta todo el código (no paso a paso) y observarás que la ejecución se para en esta instrucción saltándose el bucle que hemos puesto. En este punto. ¿Qué valor tiene \$t3 (expresado en hexadecimal y también en decimal)? ¿y qué valor tiene \$t6?

**Esta práctica se realiza en el Centro de Cálculo y no requiere entrega.**