

Práctica 3. Análisis de la ISA MIPS

En esta práctica repasaremos cómo se almacenan los datos en la memoria del computador y también veremos gráficamente cómo se ejecuta cada instrucción en el hardware del procesador.

1. Una de las características de un procesador es el llamado *endianness* que indica en qué orden se guardan en memoria los bytes individuales de datos más largos (semipalabra o palabra, por ejemplo). Si el procesador sigue el criterio *little-endian* el byte de orden más bajo se almacena en la dirección más baja. Por el contrario si el procesador sigue el criterio *big-endian* se almacenará el byte de orden más bajo en la dirección más alta.
 - a) Escriba un programa en lenguaje ensamblador MIPS que nos informe por pantalla del criterio que emplea el procesador para almacenar la información: *little-endian* o *big-endian*. ¿Se puede ver esta característica del procesador analizando la información que las ventanas del simulador ofrecen?
 - b) Pruebe el citado programa tanto en el simulador *QtSPIM* como en el *mars*. ¿Dan ambos el mismo resultado?
 - c) Pruebe almacenar un número negativo en una variable de memoria ¿en qué byte se almacena el bit de signo: en el de dirección más alta o en el de dirección más baja? Escriba un programa que, supuesto conocido el *endianness* del computador, nos diga si un entero de 32 bits almacenado en una posición de memoria es positivo o negativo analizando solo un byte.
 - d) Repita el apartado anterior con un número real representado en punto flotante tanto en simple como en doble precisión. Analice dónde se guardan los campos de signo, exponente y mantisa. De forma similar al apartado anterior, escriba un programa que, supuesto conocido el *endianness* del computador, nos diga si un número representado en punto flotante de simple precisión almacenado en una posición de memoria es positivo o negativo analizando solo un byte.
2.
 - a) Escriba un programa en ensamblador MIPS que declare una matriz bidimensional de 4x4 elementos enteros de 32 bits con los valores $a_{ij} = (-1)^i(16i - j)$, los sume todos y muestre el resultado por pantalla, declarando la matriz por filas. No debe inicializar la matriz por programa sino en la propia declaración.
 - b) Modifique el programa anterior para declararla por columnas.
 - c) Para cada uno de los programas de los apartados anteriores observe en el simulador la memoria del computador para ver cómo van quedando almacenados las componentes de la matriz.
 - d) **(Entregable)** Repita el apartado a) declarando las componentes de la matriz como enteros de 8 bits y observe cómo quedan almacenados ahora en la memoria.
3. Ejecute en el simulador *mars* alguno de los programas confeccionados en los ejercicios anteriores. En la opción del menú "*Tools*" active la herramienta "*MIPS X-Ray*" para ver como funcionaría la ruta de datos del procesador MIPS si este fuera de ciclo único:
 - a) Ejecute el programa paso a paso y analice el funcionamiento de cada instrucción ¿Qué representa cada uno de los colores que aparecen en la simulación de la ejecución?
 - b) Según la representación de la ruta de datos que realiza la herramienta: ¿Cuál es el camino crítico?
 - c) ¿Es acertada dicha representación? Proponga algunos cambios para que sea más correcta.