

## Tercera Fase de MLMV

En esta tercera fase deberá emular la ejecución de un programa escrito en lenguaje de máquina de la MVML. Para esto aplicará el ciclo de instrucción en donde inicialmente el *Program Counter (PC)* tendrá la dirección de la primera instrucción del programa a ejecutar y que se encuentra en la dirección de memoria *programa* (Fase I). Una vez obtenida la instrucción a ejecutar, se decodifica para obtener el código de operación y los operandos fuentes (Fase II). Una vez identificada la operación y sus operandos deberá invocar la función correspondiente que recibirá como argumentos los valores de los operandos y devolverá en \$v0 el resultado de la operación. Finalmente se almacena el resultado en el operando destino que indique la instrucción.

Las funciones a implementar deben respetar la Convención de Responsabilidad Compartida vista en clase. Tengan en cuenta que para una implementación eficiente de este emulador se aconseja el uso de apuntadores a funciones (ejemplo el visto en el taller para la impresión de elementos de una lista genérica).

### Cómo se manejan los Registro de la MVML?

Los registros de la máquina virtual se manejarán a través de la memoria de MIPS. Para esto deberá reservar en el área de datos estático de MIPS, el espacio necesario para mantener los 32 registros de la MVML y registros propios de la MVML (PC, etc). Así que cuando una instrucción en MVML referencie al registro 2 se refiere al registro 2 almacenado en este espacio de memoria.

### Cómo se maneja el área de datos de la MVML?

Las instrucciones lw y sw acceden al área de datos de la MVML. Para esto deben reservar un espacio estático de unas 500 palabras en MARS en donde se realizarán las operaciones relacionadas con el acceso a memoria.

Cada vez que se realice un load o un store, el cálculo de la dirección se hará relativo al inicio del espacio por usted reservado

```
.data
memoria:.space 2000
```

la dirección a la que hace referencia la instrucción lw \$4, 8(\$5) es:

memoria + \$5+8

## Cómo se manejan los branches en MVML?

En el caso de las instrucciones de salto, si se cumple la condición entonces la función asociada al branch debe actualizar el valor del PC. Recuerden que el PC apunta a la siguiente instrucción, es decir una vez que se capta una instrucción para su ejecución, se debe actualizar el PC con la dirección de la siguiente instrucción.

Cuando se cumple la condición entonces el PC se actualiza con el valor

$$PC + \text{ExtSigno}(\text{Offset}) \ll 2$$

## Salida

Una vez que finalice la ejecución del programa en MVML, deberá mostrar en la salida estandar el contenido de los registros de la MVML, y solicitar al usuario el número de palabras del área de datos de MVML a ser impresas.