

Arquitectura de Computadores.

Pablo Montoya
Ingeniería de Ejecución en Computación e Informática
Universidad del Bio-Bio
20 de Marzo de 2020

- MIPS opera con el ordenamiento **Big Endian**.
- Segmento de instrucciones se inicia en la dirección 0x00400000.
- Se utiliza instrucción “**lui**” para cargar la mitad de lo que almacena registro **\$1** para ser cargado en registro **rd**, mientras los bits restantes del dato se pasan a 0. Luego se utiliza instrucción “**ori**” para añadir los datos almacenados en los registros **\$1** y **\$2** en un mismo registro. Termina con la instrucción “**lw**” para ubicar el dato en un registro que se pueda imprimir en pantalla.
- Etiqueta “**valor**” y “**id**” poseen el mismo valor (0x3c011001) ya que los datos se sobrescriben.
- El propósito de la multiplicación por 4 en los registros **\$4** y **\$5** es para que los datos cortados por la instrucción “**lui**” vuelvan a su largo original.

Los elemento del arreglo “**valor**” luego de ser multiplicados son sobrescritos con los nuevos datos de los registros **\$4** y **\$5**.

- Instrucción “**li**” carga el valor a “**rd**”. Instrucción “**la**” carga la dirección del dato en “**rd**”.
- El programa imprime por pantalla “*esto es un array*” seguido de “99”.
- Los datos son cargados en las direcciones de memoria donde gran parte de los datos pierden la mitad de sus bits los cuales se convierten en direcciones de memoria los cuales almacenan dichos bits a la mitad. El último dato se almacena en un registro aparte donde se imprime con el mensaje “*esto es un array*”.

Se realizó seguimiento a las instrucciones que se encontraban en los archivos subidos analizando el comportamiento de estos. Se encontraron dificultades principalmente en el propósito de las instrucciones como también en la interfaz del programa pero con compañeros de la asignatura más las respuestas entregadas de parte del profesor en el foro de consultas se pudo avanzar en el manejo del simulador utilizado.