



# Lenguaje Ensamblador y la arquitectura MIPS

## 1. Objetivo General

Programar una aplicación en lenguaje ensamblador con el set de instrucciones de MIPS32 usando el simulador Spim.

## 2. Objetivos Específicos

- Entender el proceso de compilación de lenguaje de alto nivel y de lenguaje ensamblador.
- Programar una aplicación funcional usando el simulador Spim desarrollada en lenguaje ensamblador.

## 3. Materiales y Herramientas Requeridas

- 1 computador con SO basado en el núcleo Linux con Spim instalado.

## 4. Práctica

**Duración:** Para la realización de esta práctica se dispone de una semana.

### 4.1. MIPS32 y Spim

MIPS32 es una arquitectura de procesador desarrollada por la empresa Mips Technologies. La documentación de MIPS32 esta disponible en: <http://www.mips.com/products/architectures/mips32/>.

Spim es un simulador de código abierto de aplicaciones desarrolladas en lenguaje ensamblador para la arquitectura MIPS32. La pagina oficial de Spim es: <http://pages.cs.wisc.edu/~larus/spim.html>

**Actividad:** Obtener y leer la documentación de Spim y de MIPS32. ¿Cómo se carga un archivo .s? ¿Cómo se simula paso a paso? ¿Cuales son las secciones de la ventana principal de Qtspim? ¿Cuál es el set de instrucciones en lenguaje ensamblador para la arquitectura MIPS32 que soporta Spim?

### 4.2. El proceso de compilación

**Actividad:** Leer el apéndice A del libro Computer Organization and Design: The Hardware/Software Interface disponible en [http://pages.cs.wisc.edu/~larus/HP\\_AppA.pdf](http://pages.cs.wisc.edu/~larus/HP_AppA.pdf) ¿Cómo es el proceso de compilación? ¿Qué es un linker? ¿Qué es un archivo objeto? Documentar esto en el informe.



### 4.3. Lenguaje ensamblador, subrutinas e interrupciones

**Actividad:** ¿Qué es el lenguaje ensamblador? ¿Qué diferencia al lenguaje ensamblador de un lenguaje de alto nivel? ¿Qué es una subrutina? ¿Qué es una interrupción? Implemente la aplicación “Hello World” en lenguaje ensamblador para el simulador SPIM con salida por consola.

### 4.4. Aplicación en lenguaje ensamblador en Spim

#### Actividad:

Para los grupos de laboratorio 3 y 1 (Martes): Implemente en lenguaje ensamblador el algoritmo de la multiplicación. Para los grupos de laboratorio 2 y 4 (Jueves): Implemente en lenguaje ensamblador el algoritmo de la división.

#### 4.4.1. Especificaciones de diseño

La aplicación debe cumplir con las siguientes especificaciones de diseño:

- Multiplicador / divisor de dos números enteros de 32 bits. (Negativos y positivos)
- Una vez ejecutada la operación, el usuario debe poder realizar la operación nuevamente.
- La captura e impresión de los datos debe ser en formato decimal. Implementaciones en código ASCII o en codificación Hex. no serán recibidas.
- La captura de los datos debe ser de la siguiente forma: Número\_A [Enter] Número\_B [Enter].
- La captura debe detectar el ingreso de un número o un carácter y tener control de errores (mensajes de alerta al usuario, reinicializar el programa).

#### 4.4.2. Bono

Se debe implementar otro algoritmo que realice la misma operación con el fin de evaluar el desempeño. Se deben presentar tablas, gráficas o ecuaciones que evidencien la comparación de rendimiento de los algoritmos.

## 5. Evaluación

En la evaluación serán considerados los siguientes elementos:

**20 %** Asistencia por medio de un quiz.

**20 %** Trabajo en el laboratorio: funcionamiento de la aplicación, instalación de las herramientas, desarrollo de las actividades.

**40 %** Informe: El informe debe estar presentado en formato IEEE elaborado en  $\text{\LaTeX}$  con las siguientes secciones:

- Título.
- Abstract.
- Index terms.
- Introducción.
- **Estructura definida por el estudiante:** Se debe documentar el proceso de diseño de los programas, que conceptos se apropiaron con el desarrollo de la guía y los problemas y dificultades que se tuvieron durante el laboratorio. No se debe anexar código en lenguaje ensamblador. En cambio, se debe incluir los diagramas de flujo de los algoritmos implementados.



- Conclusiones.
- Referencias.

20 % Sustentación.

## Referencias

- GNU Project. Documentacion de gcc para mips. Disponible en: <http://gcc.gnu.org/onlinedocs/gcc/MIPS-Options.html>
- Harris, David & Harris, Sarah. "Digital desing and computer architecture". Prentice Hall. 2003.
- Hennessy & Patterson. "Computer Organization and Design: The Hardware/Software Interface". Elsevier. Tercera edición. Apéndice A. Disponible en: [http://pages.cs.wisc.edu/~larus/HP\\_AppA.pdf](http://pages.cs.wisc.edu/~larus/HP_AppA.pdf)
- James Larus. Documentación de Spim. Disponible en: <http://pages.cs.wisc.edu/~larus/spim.html>
- Mips Technologies. Documentación de la arquitectura MIPS32. Disponible en: <http://www.mips.com/products/architectures/mips32/>