

**Fecha de asignación:** 26 de julio de 2022

**Fecha de entrega:** 23:59 horas del 21 de agosto de 2022

### 1. Objetivo

- Estudiar la arquitectura del conjunto de instrucciones MIPS de 32 bits.
- Diseñar, codificar, ensamblar, simular y depurar programas escritos en lenguaje ensamblador MIPS.
- Familiarizarse con el uso de un entorno de desarrollo de software de bajo nivel.

### 2. Descripción

En esta práctica cada equipo de trabajo debe diseñar, codificar, simular y verificar un programa escrito en lenguaje ensamblador MIPS32 que esté en capacidad de disponer en orden alfabético un listado finito de frases.

### 3. Procedimiento

Cada equipo de trabajo debe desarrollar el programa en lenguaje ensamblador MIPS empleando la herramienta MARS<sup>1</sup>. Las condiciones a las que se debe ajustar el desarrollo son las siguientes.

- El listado de frases a ordenar estará disponible en un archivo de entrada en formato de texto plano denominado `sentences.txt`. Las frases serán construidas exclusivamente con el alfabeto inglés. El separador de frases será el mismo a lo largo del listado, pero deberá ser configurable: el programa debe consultar al usuario el separador a emplear antes de analizar el archivo de entrada.
- El listado de frases será ordenado alfabéticamente, implementando un algoritmo de ordenamiento en particular, de acuerdo con el identificador de cada equipo de trabajo, como se establece en la Sección 4 de esta guía.
- El uso de la memoria del simulador debe ser óptimo, considerándola un recurso escaso al momento de almacenar el archivo de entrada y el resultado producido por el ordenamiento.

---

<sup>1</sup> Disponible en <http://courses.missouristate.edu/KenVollmar/MARS/>

- El listado de frases ordenado alfabéticamente debe ser volcado a un archivo de texto plano con nombre `output.txt`, conteniendo una frase por línea.
- La programación en bajo nivel debe ser estructurada, es decir, debe estar basada en el uso de procedimientos, de manera que la organización modular del programa facilite su comprensión y favorezca la reutilización de código a lo largo del mismo.
- El código desarrollado debe seguir la convención para el uso de registros MIPS.
- El código fuente en ensamblador debe estar comentado de manera asertiva.

Son diversos los aspectos específicos de la implementación de la solución al problema que tendrán que ser definidos por el equipo de trabajo. En todos los casos, las decisiones de diseño deben estar ampliamente explicadas y justificadas. La calidad de estas decisiones será valorada como parte de la calificación del laboratorio.

También serán valoradas la creatividad y elegancia en la programación, así como la calidad de la interacción con el usuario mediante el uso de los servicios SYSCALL. El rendimiento del programa también será tenido en cuenta, entendiéndose este como el número de instrucciones ejecutadas para completarlo.

#### 4. Asignaciones para los equipos de trabajo

De acuerdo con el identificador asignado a cada equipo de trabajo, en la Tabla 1 se fija el algoritmo de ordenamiento que debe implementar. El número de asignación para cada equipo se calcula como (*Número del equipo* % 6).

*Tabla 1. Asignaciones para los equipos de trabajo*

# de asignación	Algoritmo de ordenamiento
0	<i>Insertion</i>
1	<i>Selection</i>
2	<i>Bubble</i>
3	<i>Merge</i>
4	<i>Heap</i>
5	<i>Quick</i>

## 5. Informe

Cada equipo de trabajo debe realizar un informe escrito que incluya una descripción del proceso de implementación del algoritmo de ordenamiento (partiendo de una representación en pseudocódigo, diagrama de flujo o lo que consideren ilustrativo), una relación de los procedimientos desarrollados (con la descripción de sus entradas, salidas y funcionalidad implementada), la explicación y justificación de las decisiones de diseño tomadas durante el desarrollo de la práctica, y resultados de ejecución para diferentes archivos de entrada. El informe debe cerrarse con las observaciones y conclusiones pertinentes. Este documento debe estar en formato **PDF** y ser subido a la plataforma antes del cierre del plazo de entrega, con el código fuente en lenguaje ensamblador como adjunto. El informe tiene un peso del 30% en la calificación global de la práctica.

## 6. Sustentación

Ambos miembros del equipo de trabajo deben demostrar un dominio completo del desarrollo de la práctica (algoritmo de ordenamiento y su implementación en bajo nivel; fundamentos conceptuales de la arquitectura del conjunto de instrucciones MIPS: formatos de instrucción, modos de direccionamiento, convención de uso de registros MIPS, uso del *stack pointer*, programación basada en procedimientos; y manejo del simulador MARS), mediante un video explicativo con una duración máxima de ocho minutos, en el que intervengan de manera equitativa. La entrega del video se debe realizar como un enlace de acceso al mismo incluido en el informe, y no como un archivo adjunto. Tanto el profesor como el monitor podrán citar a algunos equipos de trabajo para que respondan preguntas que permitan aclarar o ampliar la información suministrada en el video con la sustentación. La sustentación tiene un peso del 70% en la calificación global de la práctica.