

Fecha de entrega: 23:59 horas del 08 de junio de 2025

1. Objetivo

Estudiar la arquitectura del conjunto de instrucciones MIPS de 32 bits.

Compilar programas funcionales usando la ISA MIPS.

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 los elementos de un vector.

3. Procedimiento

Cada equipo de trabajo debe desarrollar el programa en lenguaje ensamblador MIPS empleando la herramienta MARS¹. Las condiciones a las que se debe ajustar el desarrollo son las siguientes.

- a. El listado de elementos a ordenar estará disponible en un archivo de entrada en formato de texto plano denominado *lista.txt*. El separador de elementos 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.
- b. El listado de elementos será ordenado, implementando un algoritmo de ordenamiento en particular, de acuerdo con el identificador de cada equipo de trabajo, como se establece en la Sección 5 de esta guía.
- c. El listado de elementos ordenado debe ser volcado a un archivo de texto plano con nombre *ordenado.txt*. Los elementos estarán separadas por un separador que el usuario elija.

4. Requerimientos

- a. No se permite el uso de ninguna instrucción no vista en clase.
- b. Realice un pseudocódigo o diagrama de flujo que permita resolver el problema.
- c. 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

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

comprensión y favorezca la reutilización de código a lo largo del mismo. Mínimo debe tener la implementación de 2 procedimientos.

- d. El código desarrollado debe seguir la convención para el uso de registros MIPS. Debe hacer uso de la pila mínimamente una vez por procedimiento.
- e. Realice un análisis de la complejidad computacional del algoritmo empleado, y valide esta complejidad usando el MARS.
- f. El código fuente en ensamblador debe estar comentado de manera asertiva.
- g. 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.
- h. 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, entendiendo este como el número de instrucciones ejecutadas para completarlo.

5. 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 y si es ascendente o descendente. El número de asignación para cada equipo se calcula como (*Número del equipo* % 18).

Tabla 1. Asignaciones para los equipos de trabajo

Asignación	Algoritmo	Ordenamiento
0	<i>Burbuja</i>	<i>Ascendente</i>
1	<i>Inserción</i>	<i>Ascendente</i>
2	<i>Selección</i>	<i>Ascendente</i>
3	<i>Shell</i>	<i>Ascendente</i>
4	<i>merge</i>	<i>Ascendente</i>
5	<i>QuickSort</i>	<i>Ascendente</i>
6	<i>HeapSort</i>	<i>Ascendente</i>
7	<i>CombSort</i>	<i>Ascendente</i>
8	<i>CocktailSort</i>	<i>Ascendente</i>
9	<i>Burbuja</i>	<i>Descendente</i>
10	<i>Inserción</i>	<i>Descendente</i>
11	<i>Selección</i>	<i>Descendente</i>
12	<i>Shell</i>	<i>Descendente</i>
13	<i>merge</i>	<i>Descendente</i>
14	<i>QuickSort</i>	<i>Descendente</i>
15	<i>HeapSort</i>	<i>Descendente</i>
16	<i>CombSort</i>	<i>Descendente</i>
17	<i>CocktailSort</i>	<i>Descendente</i>

6. Informe

Cada equipo de trabajo debe realizar un informe escrito que incluya:

- a. una descripción del proceso de implementación del algoritmo de ordenamiento (partiendo de una representación en pseudocódigo, diagrama de flujo o lo que consideren ilustrativo),
- b. una relación de los procedimientos desarrollados (con la descripción de sus entradas, salidas y funcionalidad implementada),
- c. 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.
- d. El informe debe cerrarse con las observaciones y conclusiones pertinentes.
- e. 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 20% en la calificación global de la práctica.

Cree un archivo comprimido que incluya el reporte y los archivos de diseño de **MARS** y **cualquier otro diseño** como se describe a continuación:

- a. **Reporte:** archivo con extensión .pdf
- b. **Archivos de diseño:** .mips, .java, .pseint

El nombre del archivo comprimido debe tener el siguiente formato:

eqXX_primerNombreApellidoIntegrante1_primerNombreApellidoIntegrante2.zip.

Ejemplo: **eqXX_AnaPerez_LuisSuarez.zip.**

7. Sustentación

- Mediante un video explicativo con una duración máxima de **ocho** minutos, en el que intervengan de manera equitativa (los integrantes deben prender la cámara en el video). 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).
- Comience la sustentación mostrando la funcionalidad de 2 sets de datos. Luego explique el desarrollo, comenzando con el algoritmo y pruebas al mismo. Explique cómo lleva este algoritmo al lenguaje MIPS. Explique cómo tuvieron en cuenta los requerimientos pedidos en la sección 4 de esta guía.
- 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 80% en la calificación global de la práctica.