

Práctica 1: “Introducción al Lenguaje Ensamblador de MIPS”

1. Descargue el simulador *MARS* desde:
<https://campusvirtual.uva.es/mod/resource/view.php?id=2456561>
2. Inicie el simulador y explore brevemente las opciones de los diferentes menús.
3. En el menú **Settings** active las opciones *Show Labels Window*, *Addresses displayed in hexadecimal*, *Values displayed in hexadecimal* y *Permit extended (pseudo) instructions and formats*. Desactive las demás.
4. Como ejercicio inicial vamos a traducir a lenguaje ensamblador de MIPS un sencillo programa escrito en lenguaje de alto nivel, concretamente C:

```
void main() {  
    int i, j;  
    int temp;  
    int A[8] = {46, 32, 83, 11, 75, 27, 64, 50};  
  
    for (i = 0; i < 8; i++)  
        for (j = 0; j < 7 - i; j++) {  
            if (A[j] > A[j + 1]) {  
                temp = A[j];  
                A[j] = A[j + 1];  
                A[j + 1] = temp;  
            }  
        }  
}
```

5. Cree un nuevo fichero mediante la opción *New* del menú **File**. A continuación, teclee en el editor de texto (se recomienda no copiar y pegar) el código en lenguaje ensamblador correspondiente al programa anterior.
6. Guarde el fichero con el nombre *P1_EquipoLab* mediante la opción *Save as...* del menú **File**. El identificador “EquipoLab” será proporcionado por el profesor encargado del grupo de laboratorio.
7. Compile el programa con la opción *Assemble* del menú **Run**, pulsando la tecla F3 o haciendo clic en el icono que tiene dos herramientas cruzadas.
8. Si el programa contiene errores observará algunos mensajes que indican el tipo de error cometido. Trate de corregirlos consultando el documento **Resumen_Esamblador_MIPS32** (se recomienda no usar ChatGPT)
9. Una vez corregidos los errores, compile de nuevo el programa. El simulador cambiará automáticamente a la pestaña *Execute*. La ventana *Text Segment*, que representa la zona de memoria donde está almacenado el código, muestra de izquierda a derecha: las direcciones de memoria, su contenido (codificación de instrucciones en lenguaje máquina), las instrucciones y el programa original.
10. Ejecute el programa paso a paso con la opción *Step* del menú **Run** o pulsando la tecla F7. Observe cómo cambia el contenido de los registros en la parte derecha de la pantalla (pestaña *Registers*). En la ventana *Data Segment* puede ver la zona de memoria donde están los datos. Activando y desactivando el checkbox *Hexadecimal Values*, verá que los valores se muestran en decimal o hexadecimal.
11. A la vista del funcionamiento observado, ¿Qué operación realiza el programa? ¿Qué algoritmo implementa?
12. Seleccione ahora la opción *Reset* del menú **Run** o pulse la tecla F12. Ejecute el programa de forma continua con la opción *Go* de dicho menú o pulsando la tecla F5. ¿Por qué el programa no muestra nada en la consola (parte inferior, pestaña *Run I/O*) al finalizar la ejecución?

13. Intente optimizar el código MIPS minimizando el número de registros utilizados y disminuyendo el número de instrucciones ejecutadas.
14. **(Entregable)** Una vez optimizado el programa, modifíquelo para que copie en orden inverso en un nuevo vector B el obtenido como resultado tras la ejecución del programa. Además, el programa debe obtener la suma de las componentes del vector en el registro \$s3, la media aritmética en el registro \$s4 y la mediana en el registro \$s5.

Criterios de Entrega

- Modo de Entrega: a través de la tarea habilitada en el Campus Virtual.
- Fecha Límite: **23 de Marzo a las 23:55** horas. No se admitirán entregas fuera de plazo.
- Entregable: fichero con el código MIPS. Lo deben entregar ambos componentes del grupo y tendrá por nombre el indicado en el punto número 6.