

Tarea

El objetivo de esta tarea es que el estudiante se familiarice con el simulador

MARS Parte 1: Revisión de material e instalación de MARS

- Revise el [material disponible en Moodle](#) relacionado con el Simulador MARS
- Descargue la última versión de [MARS \(MIPS Assembler and Runtime Simulator \)](#)
- Una vez instalado MARS, existen tres opciones para su ejecución
 - **Opción A: Desktop.** Guardar el archivo jar en alguna carpeta. Ejecutar MARS haciendo doble click sobre el ícono del simulador.
 - **Opción B: Interprete de comandos DOS usando archivo jar.** Guarde el archivo jar en algún directorio. Abra un interprete de comandos DOS en dicha carpeta usando cmd. Ejecute MARS usando el comando **java -jar Mars4_5.jar**
 - **Option C: Interprete de comandos DOS usando Java classes.** Guarde el archivo jar en algún directorio. Abra un interprete de comandos DOS en dicha carpeta usando cmd. Extraiga los archivos MARS usando el comando DOS **jar -xf Mars4_5.jar**. Ejecute MARS con el comando DOS **java Mars**

Parte 2: Ambiente MARS

- Una vez que MARS esté ejecutando, abra el archivo tarea.asm seleccionando **File → Open** en la parte superior de la ventana de MARS. Esto hará que se vea en la ventana Edit de MARS el programa cargado.
- Analice el archivo y note que el editor de texto de MARS mostrará automáticamente el número de la línea de código del archivo fuente.
- Seleccione **Run → Assemble** y note el cambio en la apariencia de MARS. Analice las ventanas:
 - Text Segment
 - Data Segment
 - Labels
 - Registers

Registers

Antes de ejecutar el código ensamblado, seleccione la pestaña **Register**. Note que la mayoría de los registros están en cero excepto Stack Pointer (\$sp), Global Pointer (\$gp) y Program Counter (\$pc).

- Cuál es el contenido de estos registros en Hexadecimal y en Decimal ?

	Hexadecimal	Decimal
\$sp	0x7ffefffc	2147479548
\$gp	0x10008000	268468224
\$pc	0x00400000	4194304

Labels

La ventana **Labels** muestra todas las etiquetas o símbolos utilizados en nuestro programa escrito en lenguaje ensamblador y la dirección de memoria asociada con dicha etiqueta. Si se hace click sobre el nombre o la dirección en la tabla de símbolos se mostrará dónde se encuentra ya sea en el área de texto o de datos.

Rellene la tabla a continuación mostrando las etiquetas presentes en el programa cargado en MARS. Indique el nombre de la etiqueta y su dirección en hexadecimal. Además marque si la misma se encuentra en el segmento de texto o de datos y su contenido en memoria.

Etiqueta	Dirección (Hex)	Text / Data	Contenido (Hex)
main	0x00400000	Text	0x24020004
mensaje	0x10010000	Data	0x616c6f48
nombre	0x10010139	Data	0x67724f00
linea	0x10010155	Data	0x00000a00

Text Segment

- Cuántas instrucciones tiene el programa en lenguaje ensamblador? Nota: Sólo se consideran aquellas líneas de programa que posteriormente serán traducidas a lenguaje de máquina y colocadas en el segmento de texto.

Respuesta: 14

- En cuántas instrucciones de lenguaje de máquina fueron traducidas las instrucciones del programa en lenguaje ensamblador?

Respuesta: _____17_____

- Cuántos bytes ocupa el programa en el segmento de Texto ?

Respuesta: _____1_____

- Ubique la instrucción número 12 de su programa (li \$v0, 4). En cuál dirección de memoria se encuentra almacenada?

Respuesta: _____0x00400010_____

Cómo fue ensamblada dicha instrucción? De la respuesta en Hex

Respuesta: _____0x24020004_____

Data Segment

- Cuál es el contenido de la dirección de memoria 0x10010008? Indique su contenido en ASCII y en Hexadecimal ?

ASCII	9			2			0			1		
HEX	3		9	3		2	3		0	3		1

- Cuántos bytes ocupa el segmento de datos ?

Respuesta: _____2_____

Parte 3. Ejecución

Ejecute el programa usando **Run** → **Go**

- Cuántos registros cambiaron su contenido?

Respuesta: _____4_____

- En cuál dirección de memoria se encuentra la última instrucción ejecutada?

Respuesta:

0x00400040

Parte 4. Servicios de Entrada y Salida

MARS ofrece una forma de interactuar con el usuario recibiendo datos de entrada y/o devolviendo datos de salida. Estas entradas y salidas, así como otros servicios son ofrecidos a través de llamadas al sistema de operación usando la instrucción syscall.

Cómo es posible decirle al sistema de operación cuál servicio queremos utilizar?. La respuesta es simple. Antes de usar la instrucción syscall, es necesario cargar en diferentes registros los parámetros requeridos por el sistema para indicarle qué tipo de servicio debe ejecutar. Revise el menu de ayuda de MARS para conocer todos los servicios (SYSCALLS) disponibles y los parámetros requeridos.

Modifique tarea.asm para solicitarle al usuario su nombre y su edad y por último escriba por consola el nombre del usuario así como su edad. Permita al usuario entrar un nombre hasta de 40 caracteres.

Fecha de Entrega: viernes 26 de enero hasta las 11:55pm. Esta tarea es **INDIVIDUAL**

La entrega consiste en dos archivos que subirá al aula virtual:

- un archivo asm con el programa modificado y que identificará con su carnet (01-23456.asm)
- un pdf con las respuestas de las partes 2 y 3 (01-23456.pdf)

tarea.asm

```
.data
mensaje:  .ascii "Hola XX-XXXX"      # Sustituya XX-XXXXX con su carnet
          .space XXX                # ultimos tres digitos de su carnet
nombre:   .ascii "Organizacion del Computador"
linea:    .ascii "\n"

.text
main:
    li $v0, 4
    la $a0, mensaje
    syscall

    li $v0, 4
    la $a0, nombre
    syscall

    la $a0, linea
    syscall

    li $t0, 10
    sll $a0, $t0, 2
    li $v0, 1
    syscall

    li $v0, 10
    syscall
```