



Universidad Simón Bolívar
Departamento de Computación y TI CI-3815
Organización del Computador
Septiembre-Diciembre 2018
Prof. Angela Di Serio

Tarea 1: Introducción a Mars

**Realizado por:
Paola Castro 13-10248**

Sartenejas; Octubre de 2018

Parte 2: Ambiente MARS

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).

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

	Hexadecimal	Decimal
\$sp	0x7ffeffc	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 indique si la etiqueta se encuentra en el segmento de texto o de datos y cuál es su contenido en memoria.

Etiqueta	Dirección (Hex)	Text / Data	Contenido (Hex)
message	0x10010000	Data	0x6e656942
num	0x100100c	Data	0x0000002d
main	0x00400000	Text	0x3c011001
is_prime_test	0x0040000c	Text	0x0104482a
is_prime_loop	0x00400001c	Text	0x0088001a
is_prime_loop_cont	0x00400034	Text	0x21080001
return	0x00400003c	Text	0x00022020

Text Segment

2. 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.

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

20

4. Cuántos bytes ocupa el programa en el segmento de Texto ?

80bytes

5. Ubique la instrucción **bne \$t1, \$zero, is_prime_loop** de su programa. En cuál dirección de memoria se encuentra almacenada?

0x00400010

6. Cómo fue ensamblada dicha instrucción? De la respuesta en Hexadecimal

0x15200002

Data Segment

- 7.Cuál es el contenido de la dirección de memoria 0x10010004? Indique su contenido en ASCII y en Hexadecimal?

ASCII	i		n		e		v	
Hex	6	9	6	e	6	5	7	6

8. Cuántos bytes ocupa el segmento de datos?

16 bytes

Parte 3. Ejecución

Ejecute el programa usando **Run → Go**

9. Cuáles registros cambiaron su contenido?

\$at, \$v0, \$t0, \$t1,\$t4, \$pc, \$lo

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

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 menú de ayuda de MARS para conocer todos los servicios (SYSCALLS) disponibles y los parámetros requeridos.

Modifique el programa EsPrimo.asm para que solicite el número a verificar y escriba por consola el resultado indicando “El número es primo” o “El número no es primo” en lugar de 1 o 0.

Fecha de Entrega: lunes 15 de octubre 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 secciones 2 y 3 (01-23456.pdf)