

Mars

Mars (**M**ips **A**ssembly and **R**untime **S**imulator) es una aplicación que permite ensamblar y simular la ejecución de programas en lenguaje ensamblador MIPS. Puede ser usado a través de línea de comando o a través de su ambiente integrado de desarrollo (IDE). Mars está escrito en Java y requiere al menos la versión 1.5 de J2SE Java Runtime Environment (JRE) para su ejecución. Se distribuye como un archivo JAR ejecutable. Puede ser descargado desde la página <http://www.cs.missouristate.edu/MARS/>

1 Descripción del Ambiente

El ambiente integrado de desarrollo (IDE) es invocado cuando se ejecuta Mars a través de la línea de comandos sin especificar argumentos, es decir `java -jar mars.jar` o haciendo doble-click sobre el ícono `mars.jar` que representa el archivo JAR ejecutable. El IDE provee facilidades para edición, ensamblaje y ejecución.

En la figura 1 podemos observar como luce la pantalla inicial cuando ejecutamos Mars.

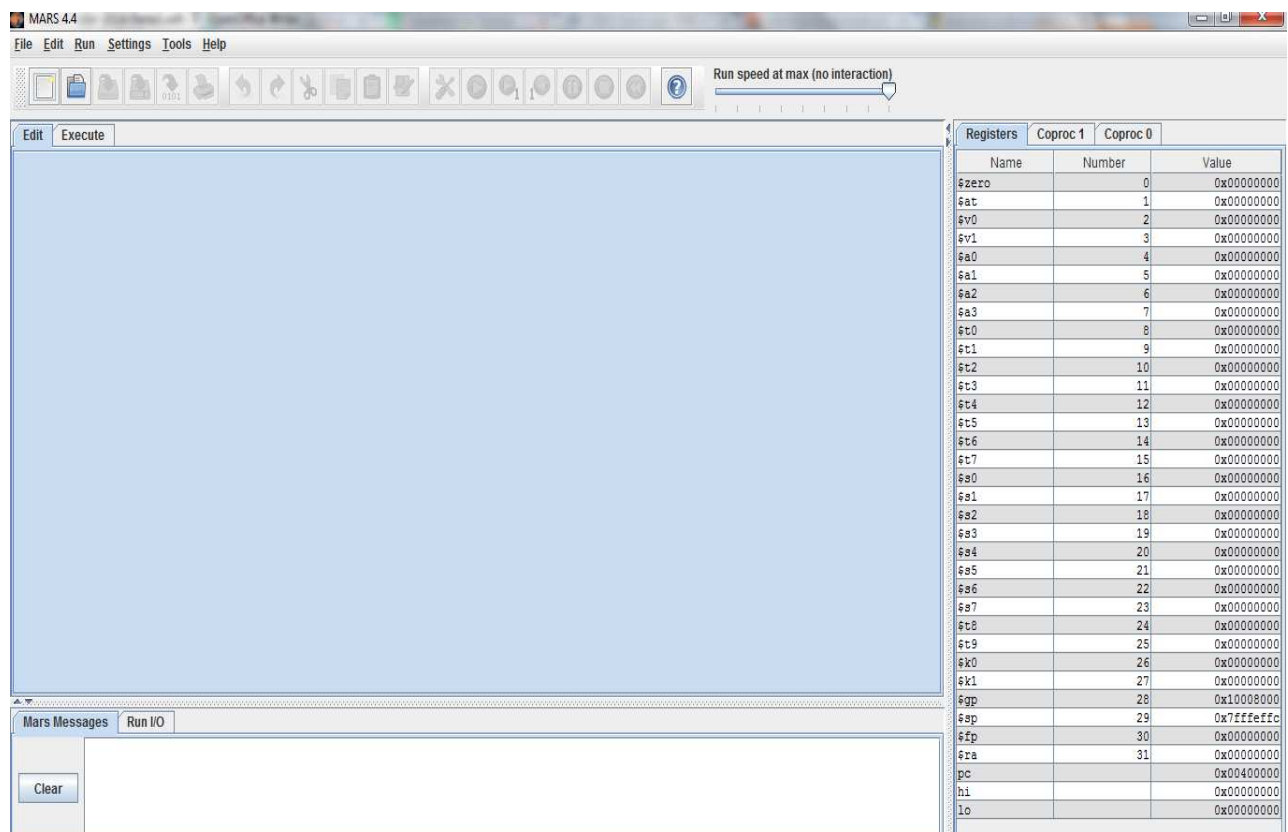


Figura 1. Pantalla inicial Mars

Entre las características que presenta tenemos:

- **Menú y Barra de Herramientas:** La gran mayoría de las opciones tienen un ícono en la barra de herramientas. Si la función de la opción no es obvia puede obtenerse una breve descripción al pasar el mouse sobre el ícono.



Figura 2. Menú y Barra de Herramientas

- **Editor:** MARS incluye un editor de texto integrado al ambiente de desarrollo. A través de “*Editor Settings*” se puede modificar el tamaño de la letra (*font*), colores, etc.

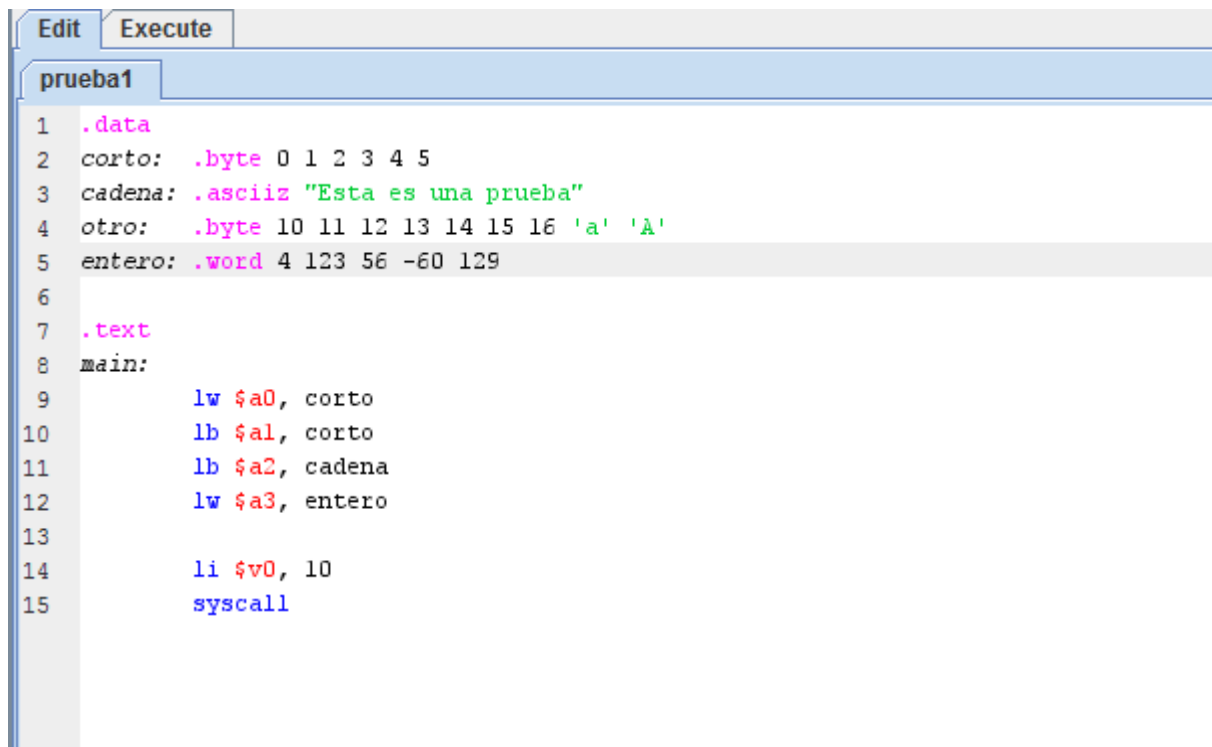


Figura 3. Editor

- **Areas de Mensajes:** En la parte inferior de la pantalla de Mars hay dos pestañas destinadas a los mensajes. La pestaña “*Run I/O*” es usada a tiempo de ejecución para mostrar mensajes de entrada y salida de datos. La pestaña “*MARS Messages*” es usada por el ambiente para mostrar errores de ensamblaje o de ejecución.

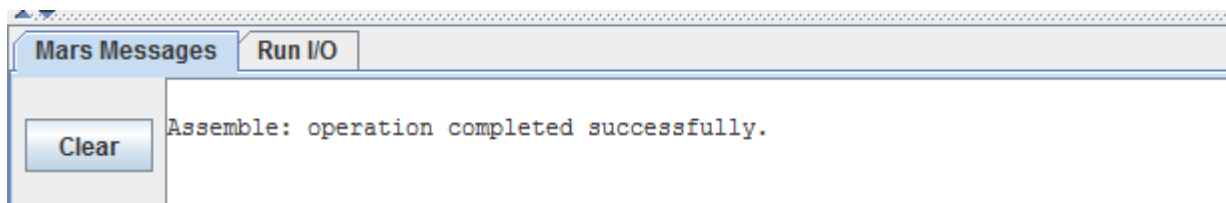


Figura 4. Area de Mensajes

- **MIPS Registers:** Los registros de MIPS se muestran todo el tiempo. Mientras se edita un programa, la ventana de registros es útil como referencia para los nombres de registros. Si se desplaza el mouse sobre los nombres de registro, se obtendrá una breve descripción sobre el uso del registro por convención. En esta ventana tenemos tres pestañas: “*Register File*” (registros de enteros \$0 al \$31, LO, HI y el Program Counter), “*Coprocessor 0*” (algunos registros para el manejo de excepciones e interrupciones), y “*Coprocessor 1*” (registros para punto flotante).

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000

Figura 5. Registros

- **Ensamblaje:** Seleccione en el menú la opción “*Run*” y después seleccione “*Assemble*” o el ícono correspondiente en la barra de herramientas para ensamblar el archivo que se encuentra activo en la pestaña “*Edit*”



- **Execution:** Una vez que el programa es ensamblado sin errores y se han inicializado los registros y los segmentos de memoria, entonces puede ejecutarse el programa a través del ícono de “*Execution*”



- **Labels Window:** Esta ventana puede o no ser mostrada. Esto será controlado a través de los “*Settings*” del ambiente. En esta ventana se muestran 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.

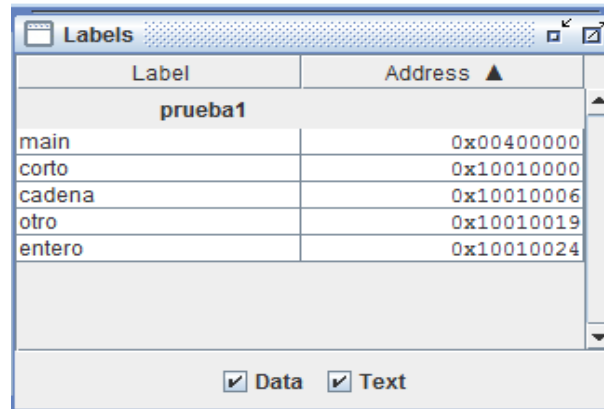


Figura 6. Ventana de símbolos

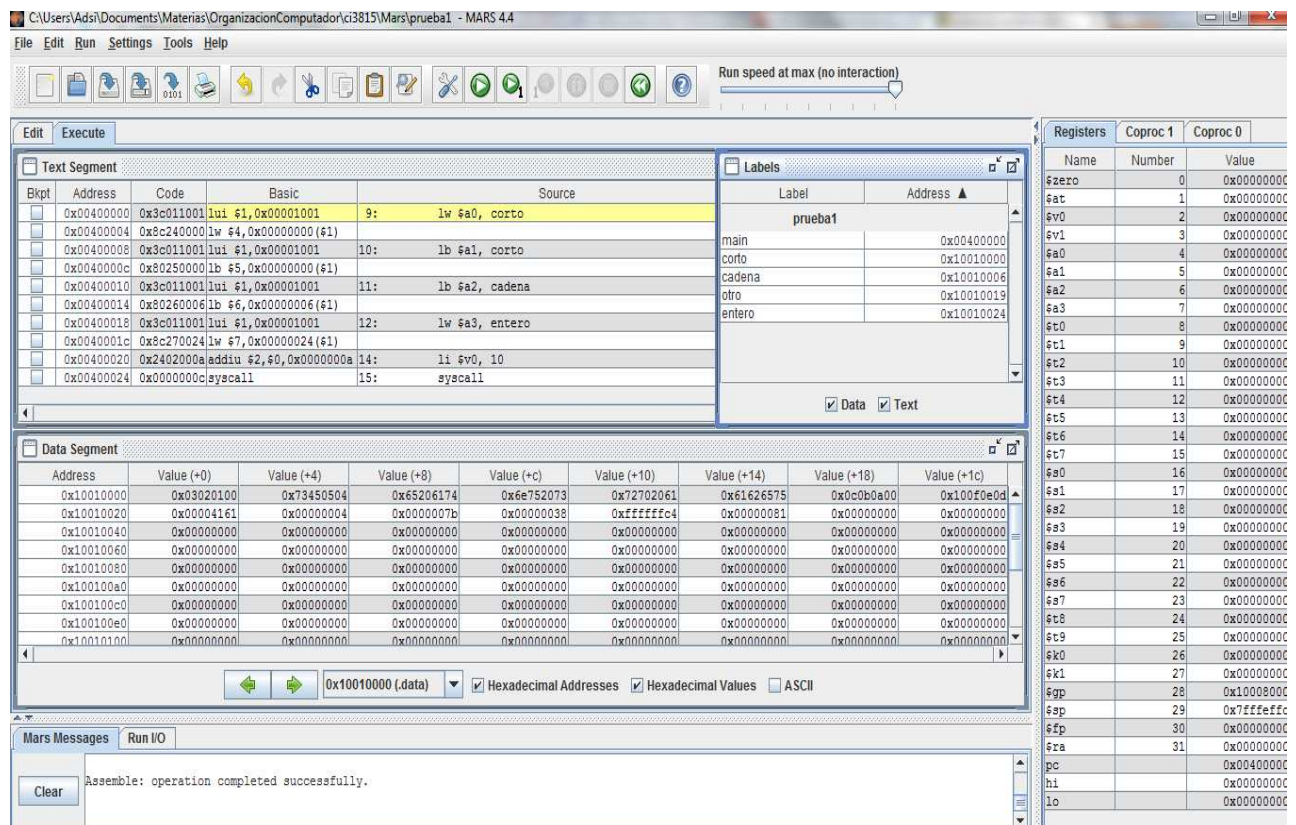


Figura 7. Pantalla de Mars con un programa ensamblado

2 Cómo se usa Mars ?

Mars permite abrir archivos con extensión .s y .asm. Para crear un nuevo archivo y poder editarlo usando la facilidad de edición, debemos hacer click en File → New, o abrir un archivo ya existente a través de File → Open.

Mars nos permite cierta interacción y la obtención de información a medida que escribimos nuestro programa (Figura 8, 9 y 10).

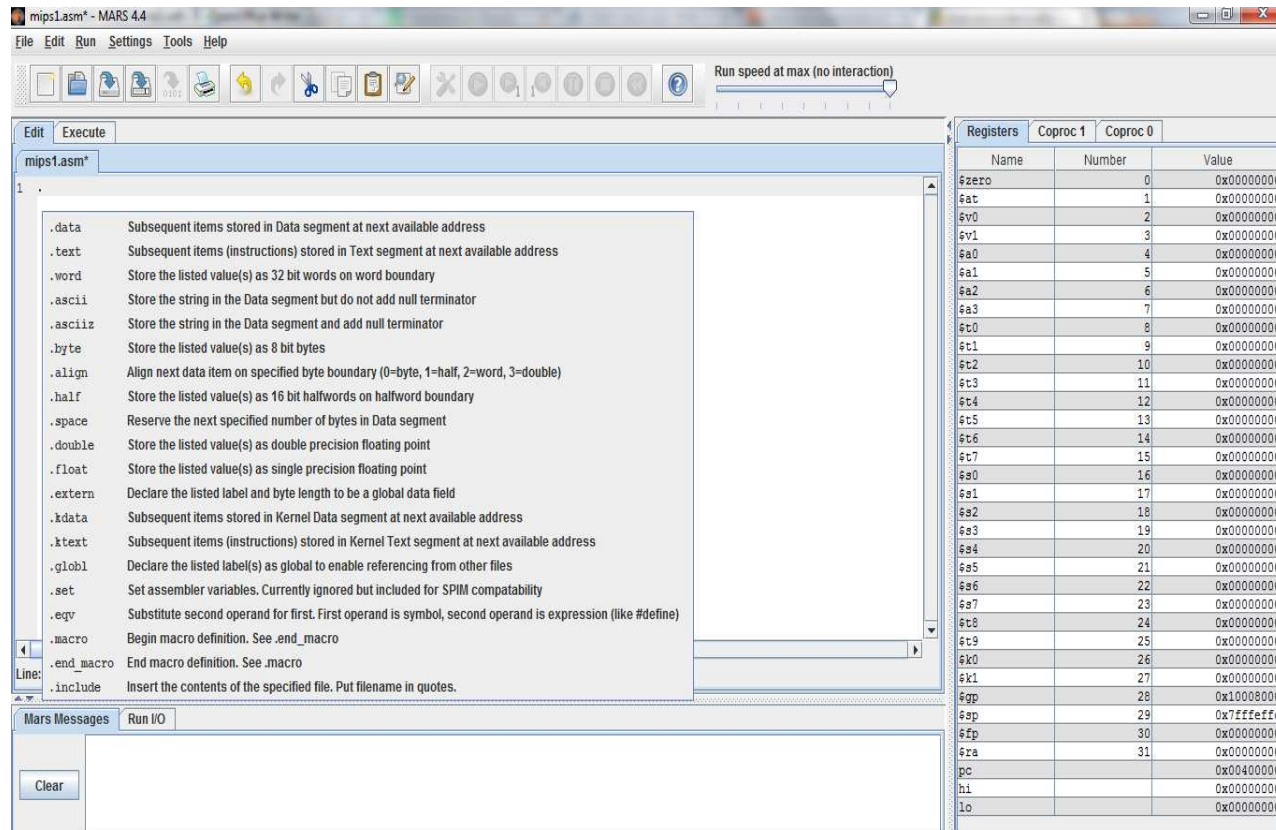


Figura 8. IDE Entorno de Desarrollo Integrado

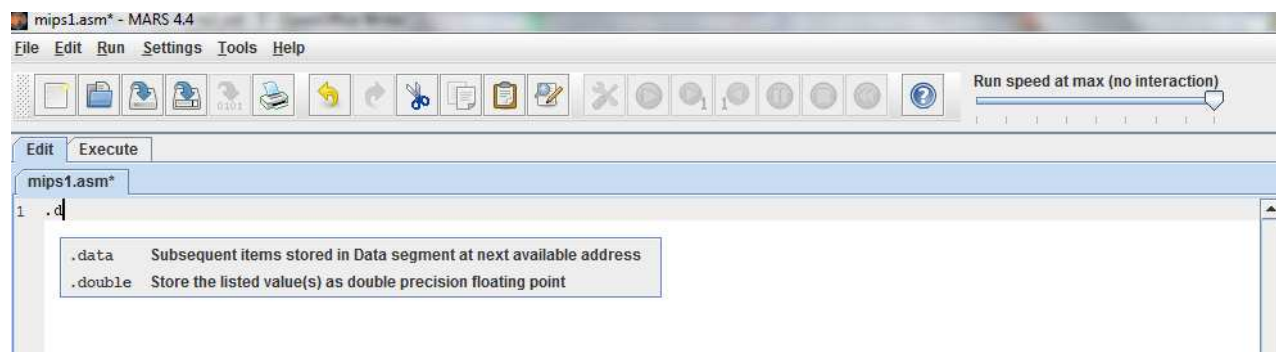


Figura 9. IDE Entorno de Desarrollo Integrado

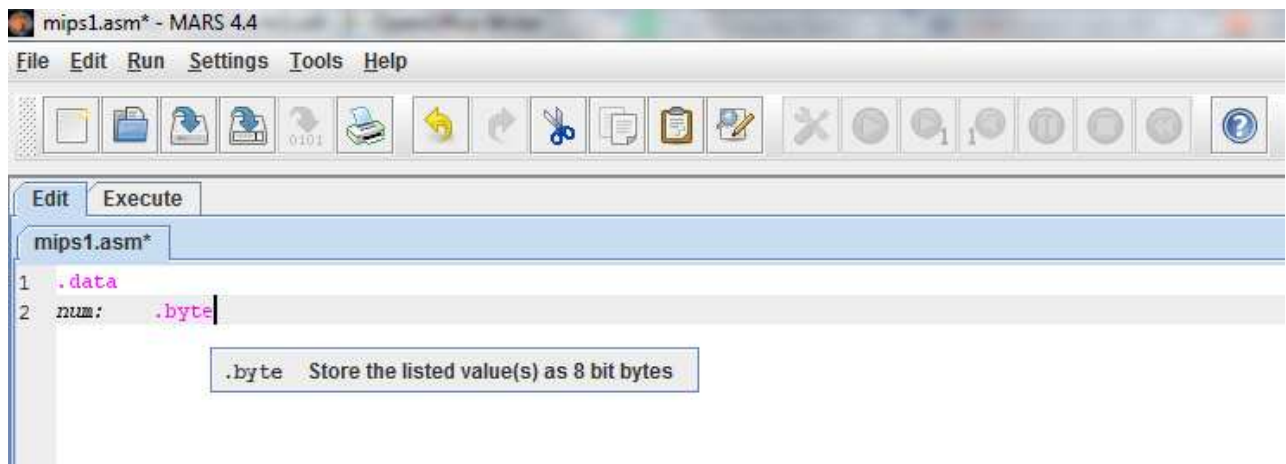


Figura 10. IDE Entorno de Desarrollo Integrado

Una vez editado nuestro programa debemos ensamblarlo para después ejecutarlo. Si se trata de un archivo nuevo, primero debe ser guardado. Para su ensamblaje, seleccionamos la opción de *Assemble*. Si hay algún error de ensamblaje aparecerán los errores en la pestaña de Mars Messages y de forma resaltada el error y la línea del programa que presenta el error (Figura 11).

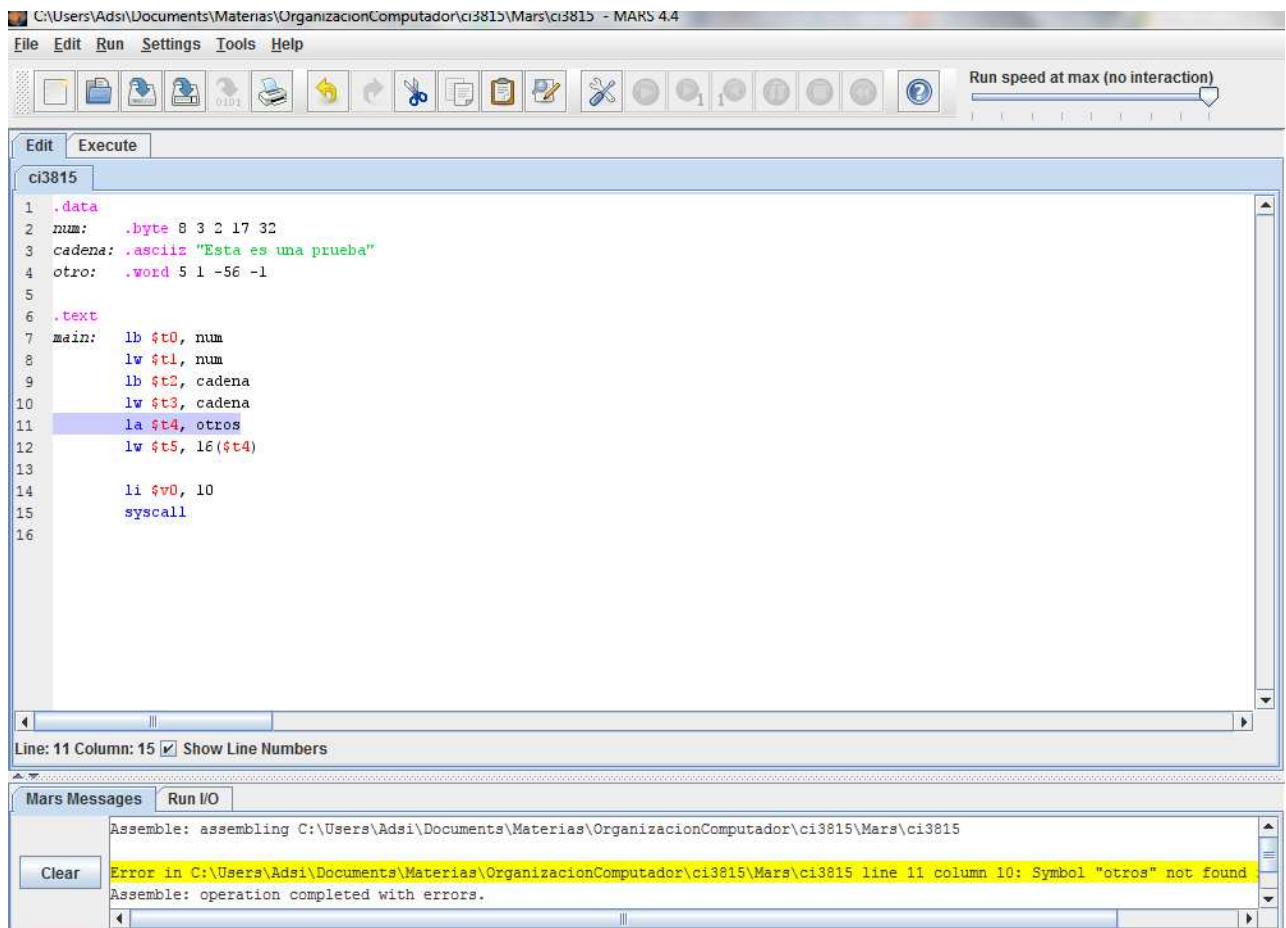


Figura 11. Ensamblaje con error

En nuestro ejemplo, el error se debe a que en el área de datos hemos declarado la etiqueta *otro* y en la línea 11 del programa colocamos *otros* que no ha sido declarada. Al corregir este error, nuestro programa es ensamblado satisfactoriamente (Figura 12).

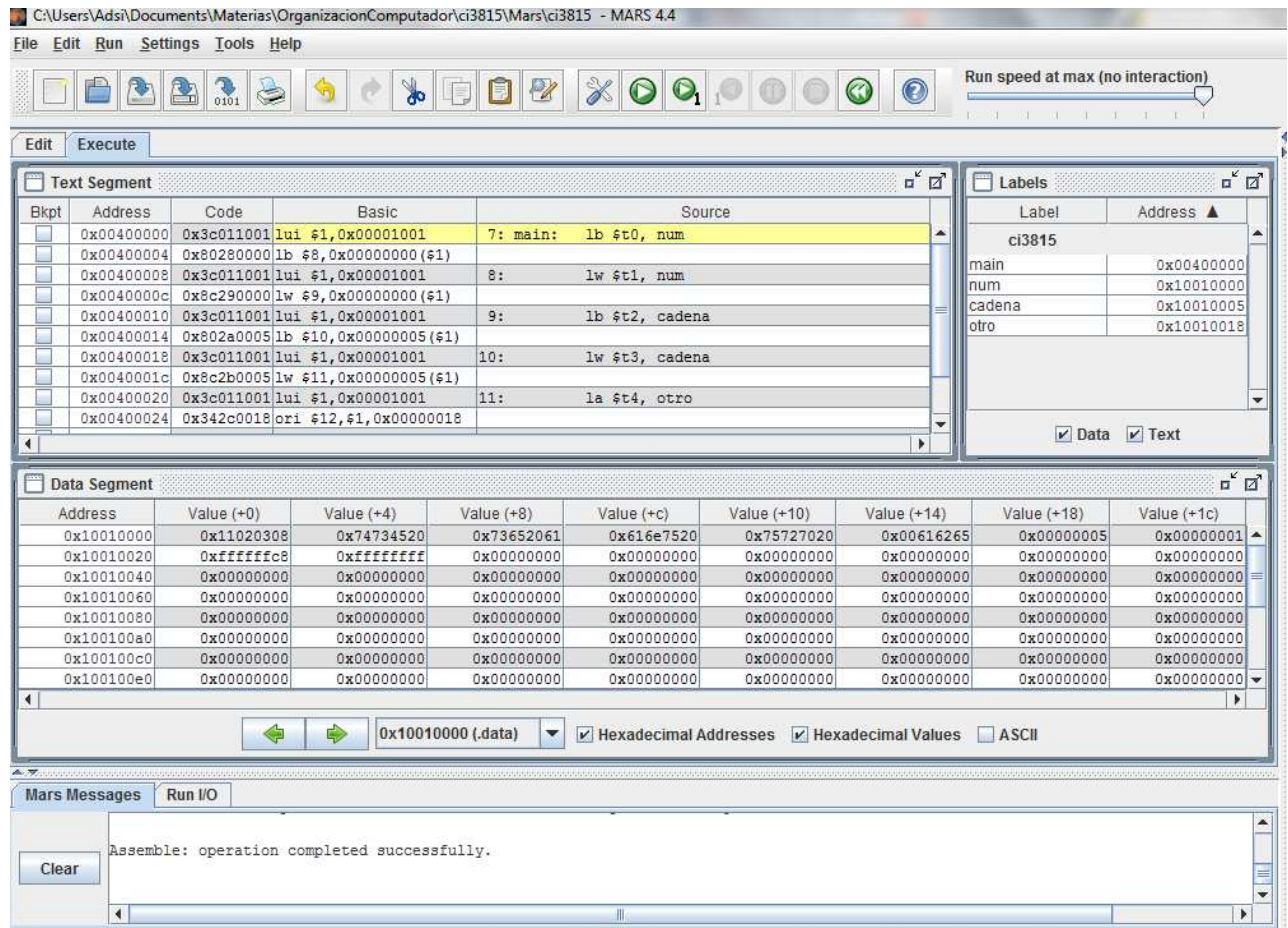


Figura 12. Ensamblaje sin errores

En la figura 12 podemos observar el contenido del área de texto. La primera columna contiene la dirección de memoria donde se almacena cada una de las instrucciones del programa (0x00400000). La segunda columna contiene cada instrucción en lenguaje de máquina. La tercera columna contiene el programa en lenguaje ensamblador que genera el ensamblador a partir de nuestro programa. Esto se debe a que el lenguaje ensamblador permite una serie de instrucciones y modos de direccionamiento (acceso a operandos) que no pertenecen al conjunto de instrucciones de la arquitectura MIPS. La última columna contiene nuestro programa. La columna identificada con Bkpt se utiliza para marcar puntos en los cuales queremos que se detenga la ejecución del programa para así revisar contenido de registros, memoria, etc. Se conocen como *breakpoints*.

Ahora podemos ejecutar nuestro programa y ver qué sucede con los contenidos de los registros.

Al ejecutarlo, lo primero que observamos es que pasa a desplegar los registros del coprocesador 0, y además notaremos que en el área de *Mars Messages* aparece el mensaje de "Go: execution terminated with errors". En este caso tenemos un error de ejecución. Qué podemos hacer para ubicar el error? (Figura 13)

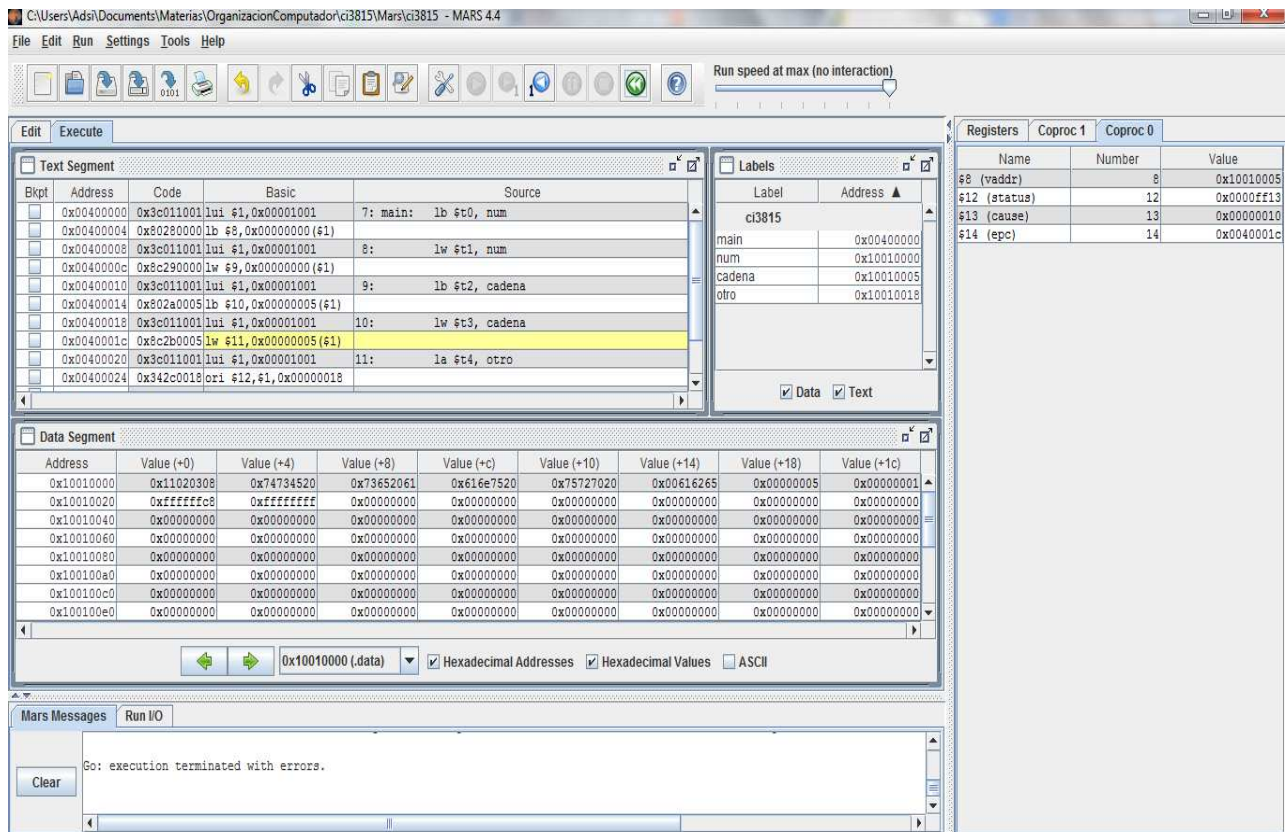


Figura 13. Ejecución con error

La herramienta nos muestra, resaltado en otro color, el punto dónde se detuvo la ejecución y la instrucción que ocasionó el problema. También podemos obtener esa información a partir del registro epc del coprocesador 0 (0x 0040001c).

El problema que se presenta en este ejemplo es muy común. Aún no estamos en capacidad de entenderlo pero poco a poco irán captando qué sucede. A grandes rasgos, el problema se trata de alineación de memoria. En la instrucción después de la 10 (sin número pues fue agregada por el ensamblador) estamos tratando de acceder a una palabra de memoria. Las palabras de memoria deben estar alineadas con direcciones múltiplos de 4 pues cada palabra consta de 4 bytes (1 palabra = 4 bytes, 1 byte=8 bits, 1 palabra=32 bits).

Por los momentos vamos a eliminar la instrucción `lw $t3, cadena` (Figura 14) y volvemos a ensamblar y ejecutar.

En esta oportunidad la ejecución finaliza sin errores. Dado que nuestro programa no lleva a cabo ningún tipo de interacción de entrada y salida, lo único que podemos observar son cambios en los registros (Figura 15).

C:\Users\Adsi\Documents\Materias\OrganizacionComputad

File Edit Run Settings Tools Help

Edit Execute

ci3815*

```

1 .data
2 num: .byte 8 3 2 17 32
3 cadena: .asciiz "Esta es una prueba"
4 otro: .word 5 1 -56 -1
5
6 .text
7 main: lb $t0, num
8       lw $t1, num
9       lb $t2, cadena
10
11      la $t4, otro
12      lw $t5, 16($t4)
13
14      li $v0, 10
15      syscall
16

```

Figura 14. Programa

C:\Users\Adsi\Documents\Materias\OrganizacionComputador\ci3815\Mars\ci3815 - MARS 4.4

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1, 0x000001001	7: main: lb \$t0, num
	0x00400004	0x80280000	lb \$8, 0x00000000(\$1)	
	0x00400008	0x3c011001	lui \$1, 0x000001001	8: lw \$t1, num
	0x0040000c	0x8c290000	lw \$9, 0x00000000(\$1)	
	0x00400010	0x3c011001	lui \$1, 0x000001001	9: lb \$t2, cadena
	0x00400014	0x802a0005	lb \$10, 0x00000005(\$1)	
	0x00400018	0x3c011001	lui \$1, 0x000001001	11: la \$t4, otro
	0x0040001c	0x342c0018	ori \$12, \$1, 0x00000018	
	0x00400020	0x8d8d0010	lw \$13, 0x00000010(\$12)	12: lw \$t5, 16(\$t4)
	0x00400024	0x2402000a	addiu \$2, \$0, 0x0000000a	14: li \$v0, 10

Labels

Label	Address
ci3815	0x00400000
main	0x10010000
num	0x10010000
cadena	0x10010005
otro	0x10010018

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000008
\$t1	9	0x11020308
\$t2	10	0x00000045
\$t3	11	0x00000000
\$t4	12	0x10010018
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x00000000
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0040002e
hi		0x00000000
lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x11020308	0x74734520	0x73652061	0x616e7520	0x75727020	0x00616265	0x00000005	0x00000001
0x10010020	0xffffffffc8	0xffffffff	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages

Run I/O

-- program is finished running --

Clear

Figura 15. Ejecución sin error