ALEJANDRO REYES ALBILLAR 45931406-S

PRÁCTICA 4.2 ESTRUCTURA DE LOS COMPUTADORES

2º) Se nos pide que realicemos en ensamblador un programa que lea una matriz 3x3 y que nos devuelva el resultado de su determinante.

Para ello, y utilizando el simulador MARS y los conocimientos adquiridos en clase hemos descrito el siguiente código:

.data

mensaje:.asciiz "Introduce el elemento ("

.align 2

coma:.asciiz ","

.align 2

cierre:.asciiz "): "

.align 2

saltolinea: .asciiz "\n"

.align 2

lineafinal1: .asciiz "El determinante de la matriz"

.align 2

lineafinal2:.asciiz " introducida es: "

.align 2

.data 0x10010100 # dirección de memoria donde se guardarán los datos leidos

elementomatriz: .word

.align 2

res: .space 4 #reserva memoria para guardar el resultado

.text

main: li $t0,0 #$t0=0

li $t1,2 #$t1=3

li $t2, 1 #$t2=1

li $t3, 1 #$t3=1

li $t4, 0 #$t4=0

pidefila: bgt $t0,$t1,finpidefila #si $t0>$t1 saltar finpidefila

li $v0,4 # código imprimir cadena

la $a0, saltolinea # imprime un salto de línea

syscall # Llamada al sistema

li $v0,4 # código imprimir cadena

la $a0,mensaje # $a0=dirección de cadena mensaje

syscall # Llamada al sistema

li $v0, 1 # código imprimir entero

li $a0, 0 # inicializa $a0

add $a0, $a0, $t2 # suma a $a0 $t2

syscall # Llamada al sistema

li $v0, 4 # código imprimir cadena

la $a0, coma # imprime el contenido de la cadena coma

syscall # Llamada al sistema

li $v0, 1 # código imprimir entero

li $a0, 0 # inicializa $a0

add $a0, $a0, $t3 # suma a $a0 $t3

syscall # Llamada al sistema

li $v0, 4 # código imprimir cadena

la $a0, cierre # imprime el contenido de la cadena cierre

syscall # Llamada al sistema

li $v0, 5 # código de leer entero

syscall # Llamada al sistema

sw $v0, elementomatriz($t4)# guarda cada elemento leido con 4 espacios de diferencia

addi $t4,$t4, 4 #$t4$t4+4

addi $t3,$t3, 1 #$t3=$t3+1

addi $t0,$t0, 1 #$t0=$t0+1

j pidefila #saltar a bucle

finpidefila: bgt $t2, 2,cargaregistros #si $t2>2 saltar operaciones

addi $t2, $t2,1 # suma a $t2 1

li $t0, 0 # inicializa $t0 a 0

li $t3, 1 # inicializa $t3 a 1

j pidefila # salta a bucle anterior

cargaregistros:li $s0, 0 # iniciliza el incremento $s0 a 0

lw $t0, elementomatriz($s0)# carga el primer valor de elementomatriz en $t0

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t1, elementomatriz($s0)# carga el segundo valor de elementomatriz en $t1

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t2, elementomatriz($s0)# carga el tercer valor de elementomatriz en $t2

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t3, elementomatriz($s0)# carga el cuarto valor de elementomatriz en $t3

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t4, elementomatriz($s0)# carga el quinto valor de elementomatriz en $t4

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t5, elementomatriz($s0)# carga el sexto valor de elementomatriz en $t5

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t6, elementomatriz($s0)# carga el septimo valor de elementomatriz en $t6

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t7, elementomatriz($s0)# carga el octavo valor de elementomatriz en $t7

addi $s0, $s0, 4 # aumenta el incremento en posiciones

lw $t8, elementomatriz($s0)# carga el ultimo valor de elementomatriz en $t8

j operaciones1 # salta a operaciones1

operaciones1:

mult $t0, $t4 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $t9 # guarda el valor de lo en $t9

mult $t9, $t8 # multiplica los valores almacenados en los registros $t9 y $t8 guardandolo en lo

mflo $s1 # guarda el valor de lo en $s1

mult $t3, $t7 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $t9 # guarda el valor de lo en $t9

mult $t9, $t2 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $s2 # guarda el valor de lo en $s2

mult $t1, $t5 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $t9 # guarda el valor de lo en $t9

mult $t9, $t6 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $s3 # guarda el valor de lo en $s3

add $s1, $s1, $s2 # suma $s1 y $s2 y lo guarda en $s1

add $s1, $s1, $s3 # suma $s1 y $s3 y lo guarda en $s1

j operaciones2 #salta a operaciones2

operaciones2:

mult $t2, $t4 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $t9 # guarda el valor de lo en $t9

mult $t9, $t6 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $s2 # guarda el valor de lo en $s2

mult $t3, $t1 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $t9 # guarda el valor de lo en $t9

mult $t9, $t8 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $s3 # guarda el valor de lo en $s3

mult $t7, $t5 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $t9 # guarda el valor de lo en $t9

mult $t9, $t0 # multiplica los valores almacenados en los registros $t0 y $t4 guardandolo en lo

mflo $s4 # guarda el valor de lo en $s4

add $s2, $s2, $s4 # suma $s2 y $s3 y lo guarda en $s2

add $s2, $s2, $s3 # suma $s2 y $s4 y lo guarda en $s2

sub $s1, $s1, $s2 # resta el valor de $s2 a $s1 guardandolo en $s1

sw $s1, res # guarda en res el valor de $s1

li $v0,4 # código imprimir cadena

la $a0,lineafinal1 # $a0=dirección de cadena lineafinal1

syscall # Llamada al sistema

li $v0,4 # código imprimir cadena

la $a0,lineafinal2 # $a0=dirección de cadena lineafinal2

syscall # Llamada al sistema

li $v0, 1 # código imprimir entero

li $a0, 0 # inicializa $a0

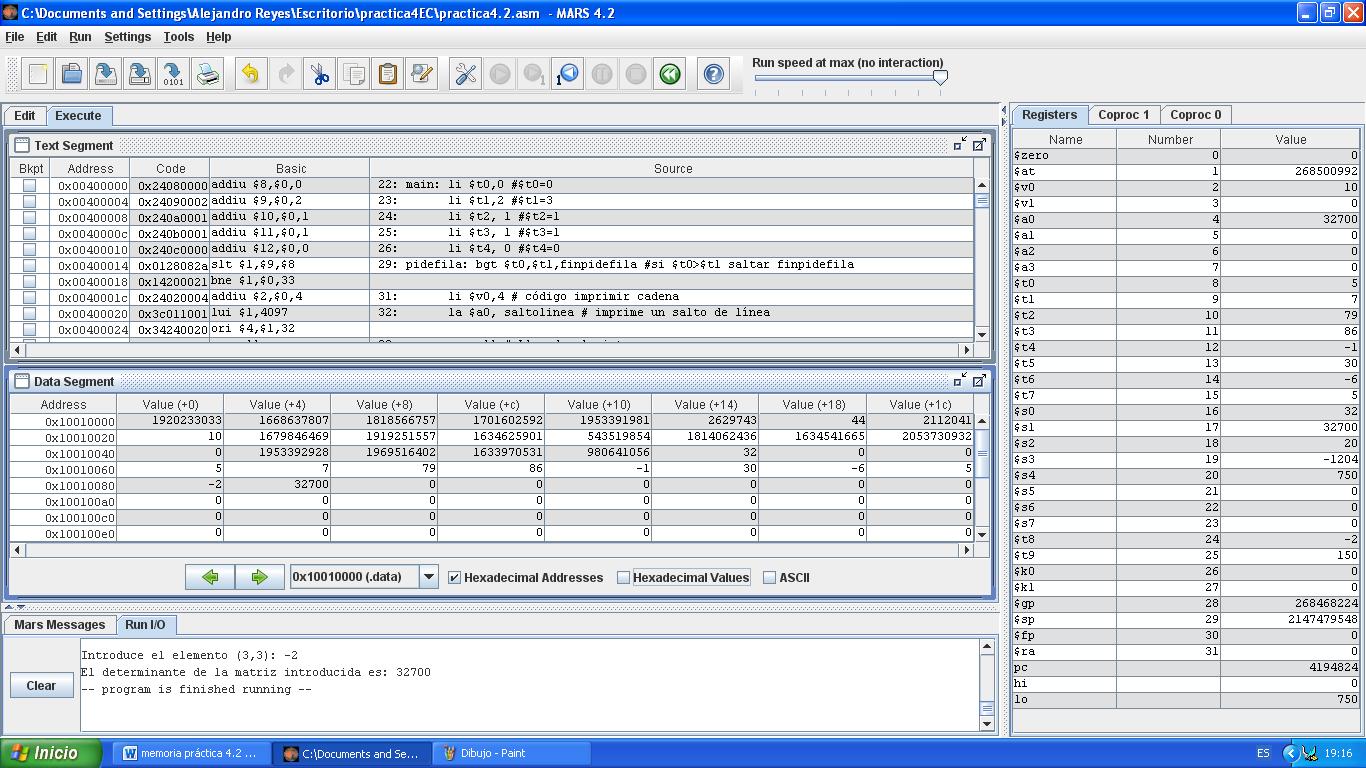
add $a0, $a0, $s1 # suma a $a0 $s1

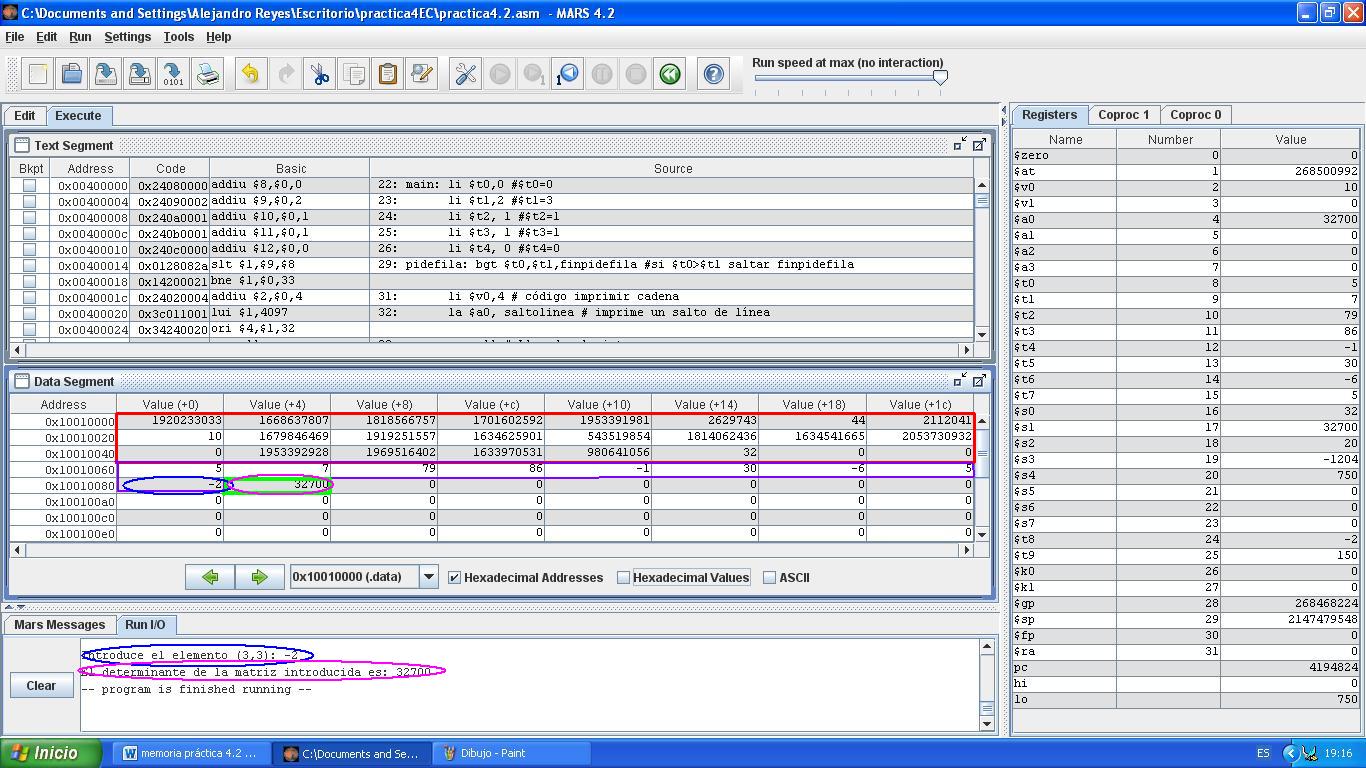
syscall # Llamada al sistema

fin: li $v0, 10 # Fin programa

syscall # Llamada al sistema

Con este código hemos leído, almacenado y utilizado los datos pedidos por consola para realizar el cálculo del determinante de la matriz.

En la siguiente imagen podemos observar los registros y la ejecución del programa con el simulador integrado en el programa del ensamblador:



Para este ejercicio hemos introducido la siguiente matriz:

Podemos observar en la imagen cómo es solicitado el último elemento de la matriz (rodeado con una elipse azul) y la línea que nos devuelve el valor del determinante de la matriz.

A partir de la dirección de memoria 0x10010060 se almacenan los nueve valores de la matriz (recuadro morado), guardándose en la posición 0x10010084 (recuadro verde) el resultado del determinante, que se nos muestra al final de la ejecución (elipse fucsia).

Las direcciones de memoria que van desde la 0x10010000 hasta la 0x10010040 (recuadro rojo) son las que guardan los diferentes campos de mensajes que se van imprimiendo según avanza la ejecución del programa.

Las imágenes anteriores son la misma con la única diferencia de las zonas resaltadas en color.

**TANTO EL CÓDIGO COMO LAS IMAGENES QUE HAN SIDO PROPORCIONADAS, SE ENCUENTRAN DENTRO DEL ARCHIVO COMPRIMIDO QUE CONTENÍA A ESTA MEMORIA.**