UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

ARQUITECTURA DEL COMPUTADOR

TALLER SOFTWARE 3

PRESENTADO POR:

LEONARDO DAVID MONSALVO CAMACHO –> T00047540

PRESENTADO A:

ING. JUAN CARLOS MARTÍNEZ SANTOS

FECHA

24/05/2019

Cartagena de Indias, Bolívar

Iniciamos la creación del código en el programa MARS4_5.jar

Tomamos el desarrollo 10.4 “Example Program, Matrix Diagonal Summation”.

Se recalca que se tuvieron que hacer cambios en multiples líneas del código porque el ejemplo dado no compilaba inmediatamente

```
#Suma de Matrices

#Impresión del resultado

#Resultado de la diagonal de la matriz resultante

#Matriz resultante en forma de fila y columna mayor



.data

MatrizA:                .word 11, 12, 13, 14, 15, 35, 56, 23

                             .word 16, 17, 18, 19, 20, 30, 23, 12

                             .word 21, 22, 23, 24, 25, 25, 34, 12

                             .word 26, 27, 28, 29, 30, 20, 34, 17
                        #Matriz de ejemplo en el libro
(MIPStextSMv11.pdf)
                             .word 31, 32, 33, 34, 35, 15, 18, 9

                             .word 11, 17, 23, 29, 35, 15, 10, 12

                             .word 14, 12, 34, 35, 11, 10, 29, 30

                             .word 12, 13, 14, 15, 16, 17, 18, 19


MatrizB:                .word 23, 43, 54, 67, 56, 5, 13, 33

                .word 11, 56, 23, 1, 4, 4, 56, 45

                .word 67, 23, 0, 44, 35, 45, 43, 23

                .word 5, 11, 23, 86, 30, 67, 22, 12  #Matriz Al azar

                .word 31, 22, 21, 35, 43, 34, 32, 34

                .word 3, 78, 55, 23, 1, 21, 32, 21, 67

                .word 7, 8, 23, 45, 34, 45, 23, 67, 54
```

```
                .word 12, 3, 80, 67, 23, 56, 45, 12, 9


Resultante:
                .word 0, 0, 0, 0, 0, 0, 0, 0
                .word 0, 0, 0, 0, 0, 0, 0, 0
                .word 0, 0, 0, 0, 0, 0, 0, 0
                .word 0, 0, 0, 0, 0, 0, 0, 0         #Matriz en 0 (
resultante )
                .word 0, 0, 0, 0, 0, 0, 0, 0
                .word 0, 0, 0, 0, 0, 0, 0, 0
                .word 0, 0, 0, 0, 0, 0, 0, 0
                .word 0, 0, 0, 0, 0, 0, 0, 0


size:           .word 8        #Tamaño de las Matrices
dSum:  .word 0


t: .asciiz "\t"
n: .asciiz "\n"
.eqv DATA_SIZE 4                              # 4 bytes for words


FilaMayor:  .asciiz "\n\t|**********Fila Mayor**********|\n"
ColumMa:    .asciiz "\n\t|**********Columna Mayor**********|\n"
matrizA:    .asciiz "\n####################|-----------Matriz A----------
|####################\n\n"
matrizB:    .asciiz "\n####################|-----------Matriz B----------
|####################\n\n"
matrizc:    .asciiz "\n####################|-----------Matriz C----------
|####################\n"


finalMsg:   .ascii    "Two-Dimensional Diagonal"
            .ascii    "Sumatoria\n\n"
```

```
            .asciiz   "Resultado de la diagonal = "

.text
.globl      main
main:
# Call function to sum the diagonal
#   (of square two-dimensional array)
     la     $a1, MatrizA          # base address of array
     lw     $a2, size                    # Tamaño de las matrices

     li $v0, 4
     la $a0, matrizA
     syscall

     jal Impresion

     li $v0, 4
     la $a0, matrizB
     syscall

     jal Impresion

     la $a1, MatrizB

     la $s1, MatrizA
     la $s2, MatrizB
     la $s3, Resultante

     jal suma
```

```
      li $v0, 4
      la $a0, matrizc
      syscall

      li $v0, 4
      la $a0, FilaMayor
      syscall

      la $a1, Resultante

      jal Impresion



      li $v0, 4
      la $a0, ColumMa
      syscall

      jal imprimirc

      jal   diagSummer
     sw $v0,dSum

#Resultado final en pantalla.
      li $v0, 4                       # print prompt string
      la $a0, finalMsg
      syscall
      li $v0, 1                       # print integer
      lw $a0, dSum
      syscall
```

```
# Se llama a la funcion para finalizar.

     li $v0, 10                # call code for terminate
     syscall                        # system call
.end main #Finaliza

# Función simple para sumar las diagonales de un
# matriz bidimensional cuadrada.

# Enfoque
# loop i = 0 to len-1
# suma = suma + mdArray [i] [i]

# Nota, para matriz bidimensional:
# addr = baseAddr + (rowIndex * colSize + colIndex) * dataSize
# Dado que la matriz bidimensional se da como un cuadrado, el
# las dimensiones de la fila y la columna son las mismas (es decir,
tamaño).

# Arguments
#     $a0 - direcciión base del arreglo
#     $a1 - size (cuadrado, arreglo bidmiensional)

# Retorna la suma de la diagonal

.globl     diagSummer
.ent diagSummer
diagSummer:
```

```
        li   $v0, 0                          # sum=0
        li   $t1, 0                           # loop index, i=0

diagSumLoop:
        mul  $t3, $t1, $a2             # (rowIndex * colSize
        add  $t3, $t3, $t1              #              + colIndex)
                                                  # note, rowIndex=colIndex
        mul $t3, $t3, DATA_SIZE   #              * dataSize
        add $t4, $a1, $t3              # + base address

        lw  $t5, ($t4)                    # get mdArray[i][i]
        add $v0, $v0, $t5            # sum = sum+mdArray[i][i]
        addi $t1, $t1, 1                # i = i + 1
        blt $t1, $a2, diagSumLoop
        # -----
# Retorna el primer valor y vuelve.
            jr  $ra
.end diagSummer

#Impresion de filas
.globl Impresion
.ent Impresion

Impresion:
                        li $t1, 0    #indice i
Impresion1:
         li $t2, 0    #indice j
Impresion2:
                        mul  $t3, $t1, $a2               # (rowIndex *
colSize
```

```
                        add  $t3, $t3, $t2                #
+ colIndex)
                                                          # note,
rowIndex=colIndex
                        mul $t3, $t3, DATA_SIZE    #              *
dataSize
           add $t4, $a1, $t3             #base address

          lw $t5, ($t4)                          #accede a M[i][j]

         li $v0, 1
         add $a0, $0, $t5
         syscall

         li $v0, 4
         la $a0, t                        #deja el espacio
         syscall

         addi $t2, $t2, 1                    #j++
         blt $t2, $a2, Impresion2

         li $v0, 4
         la $a0, n
         syscall

         addi $t1, $t1, 1                    #i++
         blt $t1, $a2, Impresion1

         jr $ra

.end Impresion
```

```
#Impresion de columnas
.globl imprimirc
.ent imprimirc

imprimirc:
                        li $t1, 0    #indice i
imprimirc1:
          li $t2, 0    #indice j
imprimirc2:
                        mul  $t3, $t2, $a2                  # (rowIndex
* colSize
                        add  $t3, $t3, $t1                   #
+ colIndex)
                                                          # note,
rowIndex=colIndex
                        mul $t3, $t3, DATA_SIZE        #              *
dataSize
          add $t4, $a1, $t3                 #base address

         lw $t5, ($t4)                              #accede a M[i][j]

         li $v0, 1
         add $a0, $0, $t5
         syscall

         li $v0, 4
         la $a0, t                          #deja el espacio
         syscall

         addi $t2, $t2, 1                   #j++
         blt $t2, $a2, imprimirc2
```

```
        li $v0, 4
        la $a0, n
        syscall

        addi $t1, $t1, 1                #i++
        blt $t1, $a2, imprimirc1

        jr $ra

.end imprimirc




# SUMA
.globl suma
.ent suma

suma:
        li $t1, 0    #indice i
suma1:
        li $t2, 0    #indice j
suma2:
                        mul  $t3, $t1, $a2              # (rowIndex *
colSize
                        add  $t3, $t3, $t2               #
+ colIndex)
                                                      # note,
rowIndex=colIndex
```

```
                        mul $t3, $t3, DATA_SIZE     #               *
dataSize

          add $t4, $s1, $t3                    #base address
          add $t5, $s2, $t3
          add $t6, $s3, $t3

          lw $t4, ($t4)
          lw $t5, ($t5)
          add $t7, $t4, $t5

          sw $t7, ($t6)                              #accede a M[i][j]

          addi $t2, $t2, 1                   #j++
          blt $t2, $a2, suma2

          addi $t1, $t1, 1                   #i++
          blt $t1, $a2, suma1

          jr $ra

.end suma
```

RESULTADOS EN PANTALLA

```
####################|----------Matriz A----------|####################

11      12      13      14      15      35      56      23
16      17      18      19      20      30      23      12
21      22      23      24      25      25      34      12
26      27      28      29      30      20      34      17
31      32      33      34      35      15      18      9
11      17      23      29      35      15      10      12
14      12      34      35      11      10      29      30
12      13      14      15      16      17      18      19

####################|----------Matriz B----------|####################

11      12      13      14      15      35      56      23
16      17      18      19      20      30      23      12
21      22      23      24      25      25      34      12
26      27      28      29      30      20      34      17
31      32      33      34      35      15      18      9
11      17      23      29      35      15      10      12
14      12      34      35      11      10      29      30
12      13      14      15      16      17      18      19

####################|----------Matriz C----------|####################

        |**********Fila Mayor**********|
34      55      67      81      71      40      69      56
27      73      41      20      24      34      79      57
88      45      23      68      60      70      77      35
31      38      51      115     60      87      56      29
62      54      54      69      78      49      50      43
14      95      78      52      36      36      42      33
81      19      42      58      56      44      74      53
79      67      26      18      96      84      41      75

        |**********Columna Mayor**********|
34      27      88      31      62      14      81      79
55      73      45      38      54      95      19      67
67      41      23      51      54      78      42      26
81      20      68      115     69      52      58      18
71      24      60      60      78      36      56      96
40      34      70      87      49      36      44      84
69      79      77      56      50      42      74      41
56      57      35      29      43      33      53      75
Two-Dimensional DiagonalSumatoria

Resultado de la diagonal = 508
-- program is finished running --
```

Los datos mostrados en pantalla son dados posteriormente a la ejecución del programa en MARS4_5.jar