

# Estructura de los computadores

## Memoria de las practicas 4 a 6

### Practica 4:

#### Cuestión 8:

Escribe una función que multiplique por 60. Escribe el programa principal que lea una cantidad de minutos y devuelve por consola la cantidad en segundos.

```
.text
li $a0,'>'
li $v0,11
syscall                                #Escribe '>'
li $v0,5
syscall                                #Lee numero

move $a0,$v0                           #Mover el numero que has introducido a la posición a0
jal mult60                             #salto a etiqueta mult60
move $a0,$v0                           #Mover el valor de v0 a a0
jal imprim                             #salto a etiqueta imprim
li $v0,10
syscall                                #fin programa

imprim:  addi $v0,$0,1
          syscall                                #imprime el resultado
          li $a0,'\n'
          li $v0,11
          syscall                                #salto de linea
          jr $ra                                #vuelve a la línea siguiente después del salto a etiqueta imprim

mult60:  sll $v0,$a0,6                      #desplazamiento lógico de 6 bits a la izquierda
          sub $v0,$v0,$a0
          sub $v0,$v0,$a0
          sub $v0,$v0,$a0
          sub $v0,$v0,$a0                  #Restamos al resultado el numero 4 veces

          jr $ra                            # vuelve a la línea siguiente después del salto a etiqueta mult60
```

#### Cuestión 9:

Modifica el código de tal manera que ahora lo que lea sea una cantidad de hora y muestre por consola la cantidad de segundos.

El código queda exactamente igual, solo agregamos estas dos líneas para que se vuelva a multiplicar por 60

```
jal mult60
move $a0,$v0
```

Quedaría así:

```
.text
li $a0,'>'
li $v0,11
syscall                                #Escribe '>'
li $v0,5
```

syscall	#Lee numero
move \$a0,\$v0	
jal mult60	#pasar de horas a minutos
move \$a0,\$v0	
jal mult60	#pasar de minutos a segundos
move \$a0,\$v0	
jal imprim	#salto a etiqueta imprim
li \$v0,10	
syscall	#fin programa
imprim: addi \$v0,\$0,1	
syscall	#imprime el resultado
li \$a0,'\n'	
li \$v0,11	
syscall	#salto de linea
jr \$ra	#vuelve a la línea siguiente después del salto a etiqueta imprim
mult60: sll \$v0,\$a0,6	#desplazamiento lógico de 6 bits a la izquierda
sub \$v0,\$v0,\$a0	
sub \$v0,\$v0,\$a0	
sub \$v0,\$v0,\$a0	
sub \$v0,\$v0,\$a0	#Restamos al resultado el numero 4 veces
jr \$ra	# vuelve a la línea siguiente después del salto a etiqueta mult60

### **Cuestión 11:**

Escribe el código que lee el valor x y escribe por pantalla la solución de la ecuación:  $5x^2 + 2x + 3$ .

```
.text

li $v0,5
syscall          #leer numero x

li $t0,5         #t0=5

mult $v0,$v0     #multiplicar x por x

mflo $t1        #mover la parte baja de la multiplicación a t1

mult $t0,$t1     #5 por x2

mflo $t1        #mover parte baja de la multiplicación a t1

li $t0,2         #t0=2

mult $t0,$v0     #2 por x

mflo $t2        #mover parte baja de la multiplicación a t2

add $a0,$t2,$t1  #se suman los resultados de t2 y t1 a a0

addi $a0,$a0,3   #se suma 3 al resultado

li $v0,1
syscall          #mostrar resultado por pantalla

li $v0,10
syscall          #fin programa
```

### **Practica 5:**

### **Question 9:**

Haz el código que lea dos enteros de la consola y escriba la suma y vuelva a comenzar si el resultado es distinto de 0. Es pseudocódigo sería:

(bucle do-while)

seguir:   Leer el primer valor (A)  
          Leer el segundo valor (B)  
          Imprimir A+B  
          Si (A+B) == 0 ir a acabar  
          ir a seguir

acabar:

```
.text

start:                #comienzo bucle

li $v0,5
syscall               #leer numero A

move $t0,$v0          #mover numero A a registro v0

li $v0,5
syscall               #leer numero B

add $a0,$v0,$t0       #sumar A y B

move $t1,$a0          #mover la suma a t1 para guardar el valor

li $v0, 1
syscall               #mostrar suma

li $a0,'\n'
li $v0,11
syscall               #salto de linea

beq $t1,$zero,end     #si suma es igual a cero fin de bucle

j start               #volver al principio del bucle

end:                  #fin de programa
```

### **Cuestión 10:**

Haz el código que lee de teclado dos valores positivos A y B en los que A<B. El programa tiene que escribir por consola los valores comprendidos entre ambos, incluyéndolos a ellos mismos. Es decir, si A=3 y B=6, escribe en la consola 3 4 5 6 (puedes escribir, por ejemplo, un salto de línea después de cada uno de los valores a mostrar).

```
.text

li $v0,5
syscall               #leer numero A

move $t0,$v0          #mover num A a t0

li $v0,5
syscall               #leer numero B

move $t1,$v0          #mover num B a t1

bucle:                #inicio bucle

sle $t2,$t0,$t1
beqz $t2,fin          #comprueba que t0(A) es menor o igual que t1(B), sino va a fin

move $a0,$t0          #mover num A a a0
```

```
li $v0,1
syscall                                #mostrar siguiente valor

li $a0,' '
li $v0,11
syscall                                #escribe un espacio en blanco

addi $t0,$t0,1                         #sumar 1 a num A

j bucle                                #salto a inicio de bucle

fin:                                    #fin programa
```

## Practica 6:

### Cuestión 7:

Escribe el código que lee cuatro enteros del teclado. Para ello deberás mostrar dos mensajes por la consola: uno que pida al usuario que introduzca el primer valor y tras haberlo leído que muestre otro solicitando el segundo valor. Los datos se almacenarán en posiciones consecutivas de la memoria, para lo cual debes haber reservado previamente espacio en el segmento de datos con la directiva `.space`. A continuación el programa leerá los valores guardados en la memoria y los mostrará en la pantalla.

```
.data
#reservamos espacio para los 4 numeros
num: .space 4
num2: .space 4
num3: .space 4
num4: .space 4

intro: .asciiz "Introduce un numero: "

.text

li $s0,1                               #contador for1
li $s1,4                               #condicion

la $t3,num                             #almacena en t3 la direccion de memoria de la etiqueta

for1:
sle $t1,$s0,$s1
beqz $t1,fin1                          #compara que la condicion se siga cumpliendo

la $a0, intro
li $v0, 4
syscall                                #escribe la frase en etiqueta intro

li $v0,5
syscall                                #leer numero

move $t2,$v0                           #mover numero a t2

sw $t2,0($t3)                          #guardar numero en direccion almacenada en t3

addi $t3,$t3,4                          #aumenta en 4 bits la direccion en t3

addi $s0,$s0,1                          #aumenta el contador

j for1                                  #salto a inicio for1

fin1:
```

```
li $s0,1                #reinicia el contador
la $t3,num

for2:
sle $t1,$s0,$s1
beqz $t1,fin2           #compara que la condicion se siga cumpliendo

lw $a0,0($t3)           #carga el numero guardado en direccion t3 a a0
li $v0,1
syscall                #imprime el numero

li $a0,'\n'
li $v0,11
syscall                #salto de linea

addi $t3,$t3,4          #aumenta la direccion en 4
addi $s0,$s0,1          #aumenta el contador

j for2
fin2:
li $v0,10
syscall                #fin programa
```

### **Cuestión 8:**

**Modifica el programa de la cuestión 7 para que muestre en la pantalla los datos guardados en la memoria ordenados de menor a mayor valor.**

```
.data
#reservamos espacio para los 4 numeros
num: .space 4
num2: .space 4
num3: .space 4
num4: .space 4

intro: .asciiz "Introduce un numero: "

.text

li $s0,1                #contador for1
li $s1,4                #condicion

la $t3,num              #almacena en t3 la direccion de memoria de la etiqueta

for1:
sle $t1,$s0,$s1
beqz $t1,fin1           #compara que la condicion se siga cumpliendo

la $a0, intro
li $v0, 4
syscall                #escribe la frase en etiqueta intro

li $v0,5
syscall                #leer numero

move $t2,$v0            #mover numero a t2

sw $t2,0($t3)           #guardar numero en direccion almacenada en t3

addi $t3,$t3,4          #aumenta en 4 bits la direccion en t3

addi $s0,$s0,1          #aumenta el contador
```

```
j for1                                #salto a inicio for1

fin1:

li $s0,1                             #reinicia el contador
la $t3,num

ordenar:
sle $t1,$s0,$s1
beqz $t1,finOrdenar                 #compara que la condicion se siga cumpliendo
li $s2,1                             #contador 2 para comparar

lw $t4,0($t3)
addi $t7,$t3,0                      #t7 es la direccion de registro del numero t4
addi $s0,$s0,1                      #aumenta el contador
j comparar
finComparar:
j ordenar

comparar:
sle $t1,$s2,$s1
beqz $t1,finComparar               #compara que la condicion se siga cumpliendo

addi $t3,$t3,4
lw $t5,0($t3)
sle $t6,$t4,$t5
beqz $t6,bubble
finbubble:
addi $s2,$s2,1                      #aumenta el contador

j comparar

finOrdenar:
li $s0,1                             #reinicia el contador
la $t3,num

for2:
sle $t1,$s0,$s1
beqz $t1,fin2                       #compara que la condicion se siga cumpliendo

lw $a0,0($t3)                       #carga el numero guardado en direccion t3 a a0
li $v0,1
syscall                             #imprime el numero

li $a0,'\n'
li $v0,11
syscall                             #salto de linea

addi $t3,$t3,4                      #aumenta la direccion en 4
addi $s0,$s0,1                      #aumenta el contador

j for2

bubble:
sw $t5,0($t7)                       #guardamos el nuevo minimo en la direccion del primero
sw $t4,0($t3)                       #guardamos el primer numero en la direccion del nuevo minimo
li $s1,4                             #para marcar fin del bucle comparar
j finbubble

fin2:
li $v0,10
syscall                             #fin programa
```