

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 1 de 5

### Relación de ejercicios

#### Tema 2: Instrucciones: lenguaje máquina. Repertorio de instrucciones MIPS

---

#### Ejercicio 1

Realizar la traducción de los siguientes fragmentos de código en lenguaje C a instrucciones MIPS (sin emplear pseudoinstrucciones) utilizando el menor número de instrucciones posible en cada caso. Las variables f, g, h, e i se han asignado a los registros \$s0, \$s1, \$s2, y \$s3 respectivamente. Las direcciones base de los vectores A y B se encuentran en los registros \$s4 y \$s5 respectivamente.

Fragmento 1	Fragmento 2	Fragmento 3
$f = g + h + i$	$f = g - h - i$  <code>sub \$s0, \$s1, \$s2 sub \$s0, \$s0, \$s3</code>	$f = g + h + B[4]$  <code>add \$s0, \$s1, \$s2 lw \$t0, 16(\$s5) add \$s0, \$s0, \$t0</code>

Fragmento 4	Fragmento 5	Fragmento 6
$f = -g + h - B[4]$	$f = g - A[B[4]]$  <code>lw \$t0, 16(\$s5) sll \$t0, \$t0, 2 add \$t0, \$s4, \$t0 lw \$t0, 0(\$t0) sub \$s0, \$s1, \$t0</code>	$f = g + A[B[h] + 1]$

#### Ejercicio 2

Realizar la traducción de los siguientes fragmentos de código de instrucciones MIPS a lenguaje C. Las variables f, g, h, e i se han asignado a los registros \$s0, \$s1, \$s2, y \$s3 respectivamente. Las direcciones base de los vectores A y B se encuentran en los registros \$s4 y \$s5 respectivamente.

Fragmento 1	Fragmento 2	Fragmento 3	Fragmento 4
<code>add \$s0, \$s1, \$s2 sub \$s0, \$s0, \$s3</code>	<code>lw \$t0, 8(\$s5) add \$t0, \$s0, \$t0 sw \$t0, 4(\$s4)  A[1] = f + B[2]</code>	<code>lw \$t0, 20(\$s5) add \$s0, \$s1, \$t0 lw \$t0, 16(\$s4) sub \$s0, \$t0, \$s0</code>	<code>add \$t0, \$s2, \$s3 sll \$t0, \$t0, 2 add \$s0, \$s1, \$t0 sll \$t0, \$s3, 3 add \$t0, \$s4, \$t0 lw \$t0, 0(\$t0) sub \$s0, \$s0, \$t0  f = g + 4 * (h + i) - A[2 * i]</code>

#### Ejercicio 3

Realizar la codificación en lenguaje máquina de las siguientes instrucciones MIPS.

Instrucción 1	Instrucción 2	Instrucción 3	Instrucción 4
<code>add \$t0, \$t0, \$zero</code>	<code>and \$t1, \$s0, \$s1</code>	<code>lw \$t1, 4(\$s3)</code>	<code>lui \$t0, 255</code>

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

### Relación de ejercicios

#### Tema 2: Instrucciones: lenguaje máquina. Repertorio de instrucciones MIPS

---

Instrucción 5	Instrucción 6	Instrucción 7	Instrucción 8
sw \$s4, 8(\$t0)	sll \$t2, \$s0, 4	bne \$t0, \$s5, X X = tres instrucciones por debajo	j 80000

**Instrucción 1:** add \$t0, \$t0, \$zero

Se trata de una instrucción cuyo formato es de tipo R. Su codificación es:

op                    rs                    rt                    rd                    shamt                    func
000000            01000            00000            01000            00000            100000

**Instrucción 7:** bne \$t0, \$s5, X

Se trata de una instrucción cuyo formato es de tipo I. Su codificación es:

op                    rs                    rt                    imm
000101            01000            10101            00000000000000010

El valor del campo imm es 2 puesto que hay que saltar a tres posiciones por debajo de la instrucción bne, pero en el momento de sumar este desplazamiento al PC éste ya se habrá incrementado para apuntar a la siguiente instrucción (PC+4). Por tanto, sólo hay que sumar 8 (dos instrucciones, cada una de ellas de 4 bytes) al valor actual del PC, que es PC+4. Ya que todas las instrucciones ocupan 4 bytes, se puede aprovechar más el campo imm si se interpreta éste como el número de instrucciones (palabras) que han de sumarse al PC, en lugar del número de bytes. Por eso, el valor del campo imm es 2: hay que saltar dos instrucciones hacia abajo, teniendo en cuenta que el PC ya apunta una por debajo de bne.

#### Ejercicio 4

Realizar la traducción desde el lenguaje máquina a instrucciones MIPS.

Instrucción 1	Instrucción 2
0000 0001 1000 1010 1000 0000 0010 0010	1000 1101 0000 1000 1111 1111 1100 0000
Instrucción 3	Instrucción 4
0011 1100 0000 1001 0000 0100 1010 0011	1010 1110 0000 1011 1111 1111 1111 1100
Instrucción 5	Instrucción 6
0000 0010 0001 0001 0100 0000 0010 1010	0000 0000 0001 0001 0100 1001 0000 0010
Instrucción 7	Instrucción 8
0001 0010 0001 0001 0000 0000 0000 0011	0000 1000 0000 0000 0000 0010 0110 1100

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 3 de 5

### Relación de ejercicios

#### Tema 2: Instrucciones: lenguaje máquina. Repertorio de instrucciones MIPS

##### Ejercicio 5

Realizar la traducción de los siguientes fragmentos de código en lenguaje C a instrucciones MIPS tratando de hacer la implementación lo más literal posible. Las variables a e i se han asignado a los registros \$s0 y \$s1 respectivamente. La dirección base del vector F se encuentra en el registro \$s2.

Fragmento 1	Fragmento 2
<pre>a = 0; for (i=0; i&lt;5; ++i) {     a = a+i; }</pre>	<pre>a = 0; i = 5; do {     F[i--] = a++; } while (i &gt;= 0)</pre>

Fragmento 3	Fragmento 4
<pre>for (i=0; i&lt;5; ++i) {     F[i] = rst2(4*i,i); }  int rst2(int a1, int a2) {     return 2*(a1-a2); }</pre>	<pre>for (i=0; i&lt;5; ++i) {     swap(F,i,9-i); }  void swap(int v[], int a1, int a2) {     int temp;      temp = v[a1];     v[a1] = v[a2];     v[a2] = temp; }</pre>

##### **Fragmneto 1:**

```
#Inicialización de registros
move $s0, $zero # a = 0
# Operación
move $s1, $zero # i = 0
loop: add $s0, $s0, $s1
      addi $s1, $s1, 1
      slti $t0, $s1, 5
      bne $t0, $zero, loop
```

##### **Fragmneto 3:**

```
#Inicialización de registros
la $s2, F
# Operación
move $s1, $zero # i = 0
loop: sll $a0, $s1, 2
      move $a1, $s1
      jal rst2
      sll $t0, $s1, 2
      add $t0, $t0, $s2
      sw $v0, 0($t0)
      addi $s1, $s1, 1
      slti $t0, $s1, 5
      bne $t0, $zero, loop
      li $v0, 10 # Llamada al sistema para salida
```

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 4 de 5

### Relación de ejercicios

#### Tema 2: Instrucciones: lenguaje máquina. Repertorio de instrucciones MIPS

syscall

```
rst2: sub $v0, $a0, $a1  
       sll $v0, $v0, 1  
       jr $ra
```

#### Ejercicio 6

Implementar un programa mediante instrucciones MIPS que realice el cálculo de los diez primeros términos de una serie cuyo primer término es 1 y los siguientes se calculan en base a la expresión:  $a_n = 2 \cdot a_{n-1} + 1$ . Almacenar los términos en un array en memoria.

#### Ejercicio 7

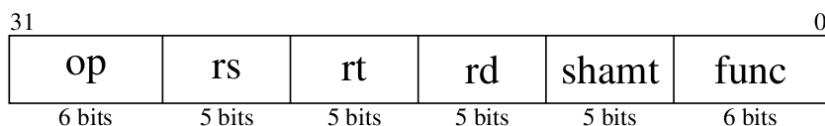
Implementar una rutina mediante instrucciones MIPS que reciba como parámetro un array de números enteros positivos y devuelva el menor y el mayor valor de los presentes en el array. Implementar un programa para comprobar su funcionamiento.

#### Material complementario

Registros del MIPS:

Números	Nombres	Uso	Preservados entre llamadas
0	\$zero	Valor constante 0	No aplicable
1	\$at	Reservado para uso temporal del ensamblador	No
2, 3	\$v0, \$v1	Resultados de las llamadas a procedimiento y evaluación de expresiones	No
4, ..., 7	\$a0, ..., \$a3	Argumentos de las llamadas a procedimiento	No
8, ..., 15	\$t0, ..., \$t7	Temporales	No
16, ..., 23	\$s0, ..., \$s7	Temporales guardados	Sí
24, 25	\$t8, \$t9	Temporales	No
26, 27	\$k0, \$k1	Reservado para su uso por el núcleo del sistema operativo	No
28	\$gp	Puntero global	Sí
29	\$sp	Puntero de pila	Sí
30	\$fp	Puntero de marco	Sí
31	\$ra	Dirección de retorno	Sí

Formato de instrucción tipo R:



# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

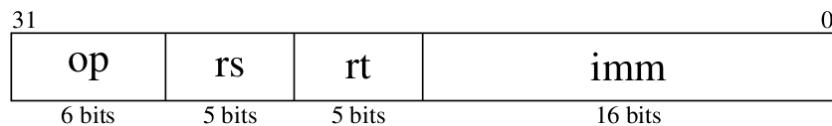
Página 5 de 5

### Relación de ejercicios

#### Tema 2: Instrucciones: lenguaje máquina. Repertorio de instrucciones MIPS

---

Formato de instrucción tipo I:



Formato de instrucción tipo J:

