Certamen 2 Arquitectura de Computadores

Nombre: Respuestas propuestas **Semestre 2019/2**

1. **Defina con sus palabras los siguientes conceptos (18 pts)**

(3 pts cada respuesta)

a. Segmento de Instrucciones:

Parte de la memoria que almacena un programa. Esta memoria es de solo lectura

(3 pts)

b. Segmento de Datos:

Parte de la memoria que permite almacenar datos. Esta memoria es de lectura/escritura

(3 pts)

c. Registro de dirección de retorno ($RA):

Este registro guarda la dirección de memoria de retorno del segmento de texto (instrucciones) cuando se invoca a una función

(3 pts)

d. Stack o Pila:

Parte del segmento de datos que permite almacenar temporalmente datos y que es gestionada de forma LiFO.

(3 pts)

e. Dirección de memoria:

Es la ubicación unica única de un byte (o palabra) en la memoria (de datos o instrucciones)

(3 pts)

f. Registro contador de programa ($PC):

Registra la dirección de memoria de la instrucción que esta en ejecución

(3 pts)

2. Considere el código assembler MIPS que es similar (no igual) al 4.s analizado en el laboratorio y responda las preguntas que se detallan luego del código. (50 Pts)

Línea	Dirección (Hex)	Programa
1		#------- segmento de texto --------
2		.text
3		.globl main
4		
5	0x4400 0000	main: la $a0, x
6	0x4400 0004	la $a1, y
7	0x4400 0008	lw $a2, size
8	0x4400 000C	
9	0x4400 0010	li $v0, 0
10	0x4400 0014	li $t3,0
11	0x4400 0018	ip: bge $t3, $a2, fin
12	0x4400 001C	lw $t0,0($a0) x[i]
13	0x4400 0020	lw $t1,0($a1) y[i]
14	0x4400 0024	mul $t2,$t0,$t1 x[i]·y[i]
15	0x4400 0028	add $v0,$v0,$t2 ∑ x_i·y_i
16	0x4400 002C	addi $a0,$a0,4 } Instrucciones
17	0x4400 0030	addi $a1,$a1,4 }
18	0x4400 0034	addi $t3,$t3,1 → Cuenta, Paran
19	0x4400 0038	b ip
20	0x4400 003C	fin: move $a0 $v0
21	0x4400 0040	li $v0,1
22	0x4400 0044	syscall
23	0x4400 0048	li $v0, 10
24	0x4400 004C	syscall
25	---	
26		#------- segmento de datos --------
27		.data
28	0x2200 0000	size: .word 5
29	0x2200 0004	x: .word 1,2,3,4,5
30	0x2200 0018	y: .word 5,4,3,2,1

a. En el código, que significado poseen las directivas .data (línea 27) y .text (línea 2) (4pts)

(2pts) .data: Indica que la siguiente información debe ser ingresada en el segmento de datos

(2pts) .text: Indica que la siguiente información son instrucciones

b. ¿Cuál es la dirección de memoria de la variable "size" y de los arreglos "x" e "y"? (4pts)

dirección de memoria	etiqueta
0x 2200 0000	size
0x 22000004	x
0x 22000018	y

(1,4 pts) (1,3 pts) (1,3 pts)

e. ¿En qué dirección de memoria se inicia el programa? (4pts)

4pts En la dirección 0x 4400 0000 (main)

d. ¿Qué valor tiene el registro $PC cuando se ejecuta la línea 10? (4pts)

4 pts 0x 4400 0014 o 0x 4400 0018

e. ¿Cuál es el valor (dirección de memoria) de las etiquetas "ip" (línea 11), y "fin" (línea 20)? (4pts)

etiqueta	dirección
ip	0x 4400 0018
fin	0x 4000 003C

2pts —
2pts —

f. Explique qué hace el programa y qué valor se imprime en pantalla. (4pts)

2pts El programa multiplica los valores de igual posición de cada arreglos $Z(x_i * y_i) =$ Suma total

2pts x: 1 2 3 4 5
y: 5 4 3 2 1 => 5 + 8 + 9 + 8 + 5 = 10 + 16 + 9 = 19 + 16 = 35

g. ¿Qué dirección de memoria posee el registro $PC luego que la instrucción de la línea 11 es ejecutada y evaluada como falsa? (4pts)

4pts 0x 4400 001C

h. ¿Qué dirección de memoria posee el registro $PC luego que la instrucción de la línea 11 es ejecutada y evaluada como verdadera? (4pts)

4 pts 0x 4400 003C

i. Modifique el programa para que calcule la suma de las diferencias entre valores de igual índice (es decir: sumar x[i]-y[i]) de los arreglos "x" e "y" (4pts). *(Puede hacer la modificación en el mismo programa o bien señale los números de las líneas del código que requieren modificación y cuál sería la modificación)*

4 pts

Cambiar linea 14 por sub $t2, $t0, $t1

j. Desarrolle un programa en C equivalente. (14 Pts)

(2 pts)

```
int main(){
    int suma=0;
```

(10 pts)

```
    for(int i=0; i<n; ++i)
    {
        suma= suma + x[i]*y[i];
    }
```

(2 pts)

```
    printf("%d", suma);
    return 0;
}
```

3. **Sofanor Rodríguez ltda, empresa innovadora en sistemas computacionales, desea desarrollar un programa en MIPS que calcule las temperaturas máximas y mínimas de un posible caso de corona virus. Para facilitarle su programa, considere que el registro de temperaturas de un paciente ya se almacena en un arreglo de acuerdo con el siguiente esquema: (32 Pts)**

```
.data
temperaturas    .word  36, 37, 35, 38, 37, 39, 38
len             .word  7
```

La variable "len" señala el largo del arreglo. Los valores máximos y mínimos deben ser almacenados en los registros $v0 y $v1 de MIPS.

```
main:   li $t1, 1                   # variable auxiliar que guarda un "1"
        la $a0, temperaturas        # contiene dirección arreglo temperaturas
        la $t0, len.                # contiene dirección de memoria de len.
        add $s0, $0, $0             # indice para recorrer arreglo
        lw $s1, 0($t0)              # carga len=7 en registro $s1
        lw $v0, 0($a0)              # carga mínimo = temperatura[0]
        lw $v1, 0($a0)              # carga máximo = temperatura[0]
repetir: lw $t1, 0($a0)             # temperatura[i] = $t1 (i=0)

        beq $s0, $s1, fin           # si indice = 7 => fin
        slt $t3, $t1, $v0           # nuevo menor mínimo
        beq $t3, $1, nuevo_mínimo

volver1: slt $t3, $v1, $t1          # nuevo máximo
        beq $t3, $1, nuevo_máximo
volver2: addi $a0, $a0, 4           # (proxima dirección del siguiente valor del arreglo)
        addi $s0 $s0, 1             # actualizar indice
        j repetir
fin     j fin                       # fin del programa
```

inicialización variables para acceder al arreglo (5pts)

buscar mínimo y máximo

nuevo_mínimo (10pts)

nuevo_máximo (10pts)

(1pt)

numero_minimo: addl $80, $t1, $0 # minimo = temperaturas[i]
j volver1

(1pt)

numero_maximo: add $21, $t1, $0 # maximo = temperaturas[i]
j volver2