

Arquitectura de Computadores (543426)

Certamen No. 1 (7 de mayo de 2012)

Nombre: _____

Matrícula: _____

Prob. 1:	/10
Prob. 2:	/12
Total:	/22

Reglas: Tiempo: 80 minutos. Responda en las mismas hojas del certamen y entregue **todas** las hojas al finalizar. Escriba su nombre en todas las hojas. Se prohíbe el uso de teléfonos, PDAs y calculadoras programables. En los problemas de programación, utilice la convención estándar de uso de registros MIPS.

Prob. 1: Programación assembly MIPS (10 pts.): Comente adecuadamente su código.

Un árbol binario es una estructura de datos en que cada nodo contiene información propia del nodo más dos punteros a nodos “hijos”. Cuando no existe un hijo, el puntero tiene en valor nulo (cero). La siguiente función retorna la profundidad máxima del árbol (la cantidad de nodos entre la raíz y el nodo más lejano).

```
struct nodo {
    int dato;
    struct nodo *izq;
    struct nodo *der;
}

int profundidad (struct nodo *raiz)
{
    int prof_izq, prof_der;
    if (raiz == 0)
        return 0;
    else {
        prof_izq = profundidad (raiz->izq);
        prof_der = profundidad (raiz->der);
        if (prof_izq > prof_der)
            return 1 + prof_izq;
        else
            return 1 + prof_der;
    }
}
```

Escriba una función en assembly MIPS que implemente esta función. Los parámetros se pasan en los registros \$a0 - \$a3. La función retorna su valor en el registro \$v0. Los punteros y enteros ocupan 32 bits. Escriba un código eficiente. Utilice la convención de registros de MIPS y la estructura de marcos de activación (stack frame) vista en clases para funciones recursivas.

Nombre:

Matrícula:

PROFUNDIDAD:

```
bne $a0, $zero, ELSE1
li $v0, 0
jr $ra
```

ELSE1:

```
sw $fp, -4($sp)
addi $fp, $sp, -4
sw $ra, -4($fp)
sw $a0, -8($fp)
addi $sp, $fp, -8
lw $a0, 4($a0)
jal PROFUNDIDAD
```

```
lw $a0, -8($fp)
sw $v0, -8($fp)
lw $a0, 8($a0)
jal PROFUNDIDAD
lw $t0, -8($fp)
lw $ra, -4($fp)
addi $sp, $fp, 4
lw $fp, 0($fp)
```

```
ble $t0, $v0, ELSE2
addi $v0, $t0, 1
jr $ra
```

ELSE2:

```
addi $v0, $v0, 1
jr $ra
```

Nombre:

Matrícula:

Prob. 2: Evaluación de desempeño (12 pts.): Muestre claramente su trabajo.

Considere un programa en MIPS que está compuesto por dos secciones: una sección secuencial no paralelizable, y un lazo iterativo que es posible paralelizar ejecutando distintas iteraciones en paralelo en múltiples procesadores de una CPU multinúcleo. La sección secuencial ejecuta 200 instrucciones, y cada iteración del lazo ejecuta 50 instrucciones. El lazo itera 80 veces. La descomposición de estas instrucciones por tipo está indicada en la siguiente tabla, al igual que los ciclos por cada tipo de instrucción:

Tipo de Instrucción	Ciclos por instrucción	Secuencial	Lazo
ALU (add, sub, etc.)	1	90	15
Punto flotante	5	30	20
Load/store	30	60	10
Salto condicionales	2	20	5

- a) (3 pts.) Para un procesador completamente secuencial (1 núcleo), calcule el CPI de la sección secuencial, el CPI de una iteración del lazo y el CPI del programa.

$$\#inst. \text{ sec.} = 200$$

$$\#ciclos \text{ sec.} = 1*90 + 5*30 + 30*60 + 2*20$$

$$CPI \text{ sec.} = 10.4$$

$$\#inst. \text{ lazo} = 50$$

$$\#ciclos \text{ lazo} = 1*15 + 5*20 + 30*10 + 2*5$$

$$CPI \text{ lazo} = 8.5$$

$$\#inst. \text{ prog.} = 200 + 50*80 = 4200$$

$$\#ciclos \text{ prog.} = 10.4*200 + 8.5*50*80 = 36080$$

$$CPI \text{ prog.} = 36080/4200 = 8.59$$

Nombre: Matrícula:

- b) (3 pts.) Usando la ley de Amdahl, calcule la aceleración que se conseguiría al incorporar un sistema de memorias cache que reduce el 90% de los lw/sw a 1 ciclo y el 8% a 5 ciclos. Para ello, calcule primero el nuevo CPI promedio de las instrucciones lw/sw y en base a eso calcule la aceleración introducida a estas instrucciones. Calcule el nuevo CPI del programa.

$$\text{CPI}_{\text{lw/sw}} = 0.9 + 0.08 \cdot 5 + 0.02 \cdot 30 = 1.9$$

$$S_m = 30/1.9 = 16.1$$

$$F_m = (30 \cdot 60 + 80 \cdot 30 \cdot 10) / 36080 = 0.715$$

$$S = 1 / ((1 - 0.715) + 0.715 / 16.1)$$

$$S = 3.04$$

$$\text{CPI} = 8.59 / 3.04$$

$$\text{CPI} = 2.83$$

- c) (2 pts.) Un compilador paralelizante puede ejecutar el lazo iterativo en paralelo sobre múltiples núcleos. Asumiendo que la sección secuencial no puede ejecutarse en paralelo con las iteraciones del lazo, calcule la aceleración sobre el procesador de la parte (a) al ejecutar el programa paralelizado sobre 8 y 16 núcleos. Sugerencia: use la ley de Amdahl.

$$\text{Flazo} = 8.5 \cdot 50 \cdot 80 / 36080$$

$$\text{Flazo} = 0.942$$

$$S_8 = 1 / ((1 - 0.942) + 0.942 / 8)$$

$$S_8 = 5.7$$

$$S_{16} = 1 / ((1 - 0.942) + 0.942 / 16)$$

$$S_{16} = 8.58$$

Nombre:

Matrícula:

- d) (4 pts.) Considere ahora que la versión paralela del programa debe agregar 2 instrucciones tipo ALU y 2 instrucciones lw/sw (para sincronización) en cada iteración del lazo por cada núcleo que participa en la ejecución. Calcule la aceleración real sobre la máquina original, alcanzada **para el lazo iterativo** para 8 y 16 núcleos. Calcule la aceleración **para el programa original** para 8 y 16 núcleos.

$$\# \text{ciclos iteración original} = 1*15 + 5*20 + 30*10 + 2*5 = 425$$

$$\# \text{ciclos iteración } N \text{ núcleos} = 425 + 2*1*N + 2*30*N = 425 + 62N$$

$$S_{\text{iteración}}(8) = 0.462$$

$$S_{\text{iteración}}(16) = 0.3$$

$$S_{\text{lazo}}(8) = 8*0.462 = 3.696$$

$$S_{\text{lazo}}(16) = 16*0.3 = 4.8$$

$$S_{\text{programa}}(8) = 3.2$$

$$S_{\text{programa}}(16) = 3.93$$

- e) (Opcional, 1 pt.) Si ahora consideráramos paralelizar la máquina del punto (b) en vez de la máquina original, ¿esperaría que el impacto de las instrucciones de sincronización en la máquina (b) fuera mayor o menor que para el caso de la máquina original? Justifique.

La mayor incidencia de las instrucciones de sincronización está dada por las dos instrucciones lw/sw. Como la máquina del punto (b) tiene un CPI = 1.9 para estas instrucciones, el impacto debería ser menor.

En efecto:

$$\# \text{ciclos programa} = 36080 / 3.04 = 11868$$

$$\# \text{ciclos iteración} = 1*15 + 5*20 + 1.9*10 + 2*5 = 144$$

$$\# \text{ciclos lazo} = 144*80 = 11520$$

$$F_m = 11520 / 11868 = 0.97$$

$$S(8) = 6.6$$

$$S(16) = 11.5$$

Con sincronización:

$$\# \text{ciclos iteración} = 144 + 2N + 2*1.9N = 144 + 5.8N$$

$$S_{\text{iteración}}(8) = 0.756 \quad S_{\text{lazo}}(8) = 6.048$$

$$S_{\text{iteración}}(16) = 0.608 \quad S_{\text{lazo}}(16) = 9.728$$

$$S_{\text{programa}}(8) = 5.25$$

$$S_{\text{programa}}(16) = 7.71$$