

6) Como ya se observó anteriormente, muchas instrucciones que normalmente forman parte del repertorio de un procesador con arquitectura CISC no existen en el MIPS64. En particular, el soporte para la invocación a subrutinas es mucho más simple que el provisto en la arquitectura x86 (pero no por ello menos potente). El siguiente programa muestra un ejemplo de invocación a una subrutina.

```
.data

valor1:    .word 16
valor2:    .word 4
result:    .word 0

.text

ld $a0, valor1($0)
ld $a1, valor2($0)
jal a_la_potencia
sd $v0, result($0)
halt

a_la_potencia: daddi $v0, $0, 1
lazo:    slt $t1, $a1, $0
        bnez $t1, terminar
        daddi $a1, $a1, -1
        dmul $v0, $v0, $a0
        j lazo
terminar:    jr $ra
```

a) ¿Qué hace el programa? ¿Cómo está estructurado el código del mismo?

b) ¿Qué acciones produce la instrucción jal? ¿Y la instrucción jr?

c) ¿Qué valor se almacena en el registro \$ra? ¿Qué función cumplen los registros \$a0 y \$a1? ¿Y el registro \$v0?

d) ¿Qué sucedería si la subrutina a\_la\_potencia necesitara invocar a otra subrutina para realizar la multiplicación, por ejemplo, en lugar de usar la instrucción dmul? ¿Cómo sabría cada una de las subrutinas a qué dirección de memoria deben retornar?

## **Respuestas:**

6)

a) El programa realiza una potencia. Pasa la base y el exponente como parametros a una subrutina la cual realiza el calculo y devuelve el resultado, el cual es guardado en memoria principal.

b) La instruccion jal realiza un salto incondicional a una etiqueta (la cual indica el comiezo de una subrutina) y almacena en el registro \$ra la direccion de memoria de retorno de la subrutina (Direccion de la instruccion siguiente al jal). La instruccion jr realiza un salto incondicional a la direccion de memoria almacenada en \$ra. En conjunto ambas instrucciones nos permiten realizar un llamado a una subrutina.

c) En el registro \$ra se almacena la direccion de retorno de la subrutina (direccion de la instruccion siguiente al jal). En los registros \$a0 y \$a1 se almacenan los parametros a pasar a la subrutina. En el registro \$v0 se almacena el valor a devolver por la subrutina.

d) Si se precisa llamar a otra subrutina deberiamos salvar el valor almacenado por \$ra en la pila (en memoria principal) para luego poder recuperar ese direccion antes de retornar de la subrutina y asi saltar a la direccion de retorno adecuada.