

**EJERCICIO MÓDULO II. SIMULACIONES CON WINMIPS64**

Guarde este código en un archivo llamado Codigo.s para simular en WINMIPS64:

*; Add values from two vectors.  
; The result is stored in F6 register.*

*.data*

*X:     .double 1.00, 1.01, 1.02, 1.03, 1.04, 1.05, 1.06, 1.07, 1.08, 1.09  
       .double 2.00, 2.01, 2.02, 2.03, 2.04, 2.05, 2.06, 2.07, 2.08, 2.09  
Y:     .double 1.00, 1.01, 1.02, 1.03, 1.04, 1.05, 1.06, 1.07, 1.08, 1.09  
       .double 2.00, 2.01, 2.02, 2.03, 2.04, 2.05, 2.06, 2.07, 2.08, 2.09*

*.text*

*main: daddi r3,r0,20             ; r3 <-- Number of vector components to process  
       daddi r4,r0,0             ; r0 <-- cont  
       daddi r1,r0,X            ; r1 will be my base register X  
       daddi r2,r0,Y            ; r2 will be my base register Y*

*loop:*

*l.d f0, 0(r1)     ; F0 <-- X(i)  
    l.d f2, 0(r2)     ; F2 <-- Y(i)  
    add.d f4,f0,f2    ; F4 <-- X(i) + Y(i)  
    add.d f6,f6,f4    ; F6 <-- F6+f4*

*daddi r1,r1,8    ; X(i+1)  
    daddi r2,r2,8    ; Y(i+1)  
    daddi r4,r4,1    ; Y(i+1)  
    bne r4,r3,loop ; branch if r4!=r3*

*halt*

1. Ahora realice simulaciones sin ningún tipo de optimizaciones. Observe el resultado de las mismas.
2. Aplicando las optimizaciones realizadas en la primera clase, vuelva a simular el código y revise el resultado.
3. Aplique la técnica de desenrollado de bucles al código anterior, guarde el resultado en un fichero diferente, llamado Codigo2.s. Este fichero es el que subir en la entrega del Bloque II.
  - a. Ahora realice simulaciones sin ningún tipo de optimizaciones. Observe el resultado de las mismas.
  - b. Aplicando las optimizaciones realizadas en la primera clase, vuelva a simular el código y revise el resultado.
4. Adjunte en un .doc el código con cada paso dado en el desenrollado que subirá en la entrega junto al .s del código resultante del último paso.