

Evaluación del Trabajo de Laboratorio Número 1

Arquitectura de Computadoras

29/04/2020

Instrucciones

- La presente evaluación es individual y podrá utilizar todo el material escrito del que disponga.
 - La misma consta de 2 ejercicios en total y la duración será de 75 minutos.
 - En cada ejercicio se indica el tiempo estimado para resolverlo y el puntaje que otorga el mismo.
 - La evaluación se desarrollará en el entorno de Google Classroom.
 - La evaluación deberá ser desarrollada en el documento de Google Docs adjunto al enunciado.
 - Se deberá completar el encabezado del documento con los datos del alumno.
 - No se admitirán otros archivos o formatos a menos que sean explícitamente solicitados.
 - No se permitirá intercambiar ni prestar ningún tipo de elementos o información entre los estudiantes.
 - Para hacer sus soluciones **se recomienda enfáticamente que pruebe sus códigos en el simulador Venus** antes de enviarlos, ya que si no funciona en el simulador el puntaje se verá **significativamente** disminuido.
 - La legibilidad del código, la cantidad de comentarios y la buena presentación serán tenidas en cuenta para la calificación final.
-

Ejercicios para resolver

1) 35 Minutos - 50 Puntos (40, 10)

Escriba una subrutina en lenguaje ensamblador RV32IM que reciba la dirección de un vector de enteros en el registro **a0**, lea todos los elementos del vector hasta encontrar el valor cero, y en ese momento finalice indicando la cantidad de números pares e impares que contiene dicho vector.

- a) Realizar una subrutina y programa principal de prueba. La subrutina deberá retornar la cantidad de números pares en el registro **a1**, y la cantidad de números impares en el **a2**.
- b) Mostrar por pantalla la cantidad de números pares e impares.

2) 40 Minutos - 50 Puntos (40, 10)

Se desea implementar el código de una función de una máquina expendedora de gaseosas, a la cual se le ingresa un cierto monto de dinero, y devuelve la cantidad mínima de monedas que debe entregar para llegar a ese monto. La máquina trabaja con monedas de \$1, \$5 y \$10. Por ejemplo, si se ingresa el monto 45, se espera que la función devuelva que debe entregar 4 monedas de \$10 y 1 moneda de \$5.

- a) Escriba una subrutina en lenguaje ensamblador RV32IM que reciba un valor en pesos entero en el registro **a0**, y devuelva las cantidades mínimas de monedas necesarias en los registros **a1**, **a2** y **a3** (para las monedas de \$1, \$5 y \$10, respectivamente). Escriba también el programa principal de prueba.
- b) Indique por pantalla la cantidad de monedas de cada tipo que debe entregar.

Soluciones propuestas

1) Solución propuesta:

```

.data
vector: .word 72,57,39,48,46,88,2,91,0
cad1: .asciiz "La cantidad de numeros pares es: "
cad2: .asciiz "\nLa cantidad de numeros impares es: "

.text
.globl main

main:

    la a0, vector    # Cargo la dirección base del vector
    jal contador
    mv a4, a1         # Copio el valor de a1, porque lo uso para mostrar
    jal mostrar

    li a0, 10         # Fin del programa
    ecall             #

contador:
    mv a1, zero       # Inicializo el contador de pares en cero
    mv a2, zero       # Inicializo el contardor de impares en cero

lazo:
    lw t0, 0(a0)      # Cargo el elemento a utilizar
    beqz t0, fin      # Termino ciclo si elemento es cero
    addi a0,a0,4       # Incremento puntero a siguiente elemento
    andi t0,t0,1       # Enmascaro ultimo bit para determinar paridad
    beqz t0, par      # Salto si el numero es par
    addi a2,a2,1       # Incremento contador de impares
    j lazo            # Repito lazo para siguiente elemento
par:
    addi a1,a1,1       # Incremento contador de pares
    j lazo            # Repito lazo para siguiente elemento
fin:
    ret

mostrar:
    li a0, 4          # Muestro la cantidad de números pares
    la a1, cad1
    ecall
    li a0, 1
    mv a1, a4
    ecall

    li a0, 4          # Muestro la cantidad de números impares
    la a1, cad2
    ecall
    li a0, 1
    mv a1, a2
    ecall

    ret

```

2) Solución propuesta:

```

.data

cad3: .asciiz "\nLa cantidad de monedas de 10 pesos es: "
cad2: .asciiz "\nLa cantidad de monedas de 5 pesos es: "
cad1: .asciiz "\nLa cantidad de monedas de 1 pesos es: "

.text

```

```
.globl main

main:
    li a0, 91          # Cargo el valor a mostrar
    jal monedas
    mv a4, a1          # Copio el valor de a1, porque lo uso para mostrar
    jal mostrar

    li a0, 10          # Fin del programa
    ecall

monedas:
    li t3, 10          # Cargo un auxiliar para hacer las divisiones
    div a3, a0, t3      # Obtengo la cantidad de monedas de 10
    rem t4, a0, t3      # Obtengo el saldo restante
    li t3, 5           # Cargo el auxiliar para monedas de 5
    div a2, t4, t3      # Obtengo la cantidad de monedas de 10
    rem a1, t4, t3      # Obtengo el saldo restante, que son las monedas de 1

    ret

mostrar:
    li a0, 4           # Muestro la cantidad de monedas de 10
    la a1, cad3
    ecall
    li a0, 1
    mv a1, a3
    ecall

    li a0, 4           # Muestro la cantidad de monedas de 5
    la a1, cad2
    ecall
    li a0, 1
    mv a1, a2
    ecall

    li a0, 4           # Muestro la cantidad de monedas de 1
    la a1, cad1
    ecall
    li a0, 1
    mv a1, a4
    ecall

    ret
```