

# Organización de Computadoras

## Guía de Ejercicios Nº 4 - ASM Risc V

1. Cargar un valor en **a0** y asignarle un 10 a **s1** si es par o un 20 si es impar
2. Sin utilizar las instrucciones de multiplicación, multiplicar el valor que está en **t1** por 61 y dejar el resultado en **a0** (trate de hacer una versión con ciclos y otra con desplazamientos)
3. Realice un programa que copie el vector V de 15 palabras en otro llamado A.
4. Resuelva el problema anterior pero únicamente copiando los valores impares.
5. Codifique un programa que dados tres valores en el segmento de datos, determine el menor. El valor del menor guárdelo en memoria a continuación de los tres valores.
6. Teniendo un vector de diez palabras, recórralo y multiplique cada uno de sus valores por ocho.
7. Copiar el siguiente programa, analícelo y responda (ejecútelo paso a paso)

```
1  .data
2  cadena: .asciz "ORGANIZACION DE LAS COMPUTADORAS CON RISC V"
3          .align 2
4  cant:   .byte -1
5          .align 2
6  caract: .ascii "A"
7  .text
8          la a0,cadena
9          lb t1,caract
10         ori a1,zero,0    # contador
11 ini:     lb t0,0(a0)
12         bne t0,t1,distin
13         addi a1,a1,1
14 distin: beq t0,zero,fin
15         addi a0,a0,1      # incremento solamente 1 byte
16         j ini
17 fin:     sw a1,cant,t6
18
19         la a0,cadena
20         li a7,4
21         ecall
22
23         ori a7,zero,10
24         ecall
```

- ¿Qué se almacena en la dirección apuntada por *cant*? ¿Cuál es esta dirección?
- ¿Cuál es la finalidad del programa?
- Analice e interprete las instrucciones en ensamblador generadas por:  
`la a0, cadena      lb t1, caract      lb t0, 0(a0)      sw a1, cant, t6`
- Investigar el uso de la directiva **.align**. ¿Cuál es su función? Quítela y verifique cómo quedan organizados los datos en la memoria del segmento de datos (0x10010000 en adelante). ¿Qué entiende por alineación de memoria?
- ¿Cómo se organizaron los datos en memoria? Determine cómo está guardado el mensaje de texto.
- ¿Cuál es la dirección apuntada por la etiqueta *caract*? ¿y por *cant*?

The top screenshot shows the Data Segment window with the following data:

| Address    | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x4147524f | 0x415a494e | 0x4e4f4943 | 0x20454420 | 0x2053414c  | 0x504d4f43  | 0x44415455  | 0x5341524f  |
| 0x10010020 | 0x4e4f4320 | 0x53495220 | 0x00562043 | 0x000000ff | 0x00000041  | 0x00000000  | 0x00000000  | 0x00000000  |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000  | 0x00000000  | 0x00000000  | 0x00000000  |

The bottom screenshot shows the same Data Segment window with the 'ASCII' checkbox checked. The data is displayed in ASCII format:

| Address    | Value (+0)  | Value (+4)  | Value (+8)  | Value (+c)  | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 0x10010000 | A G R O     | A Z I N     | N O I C     | E D         | S A L       | P M O C     | D A I U     | S A R O     |
| 0x10010020 | N O C       | S I R       | \0 V C      | \0 \0 \0 .  | \0 \0 \0 A  | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010040 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |

- Modifique el programa anterior de tal manera que modifique las letras 'A' por letras 'x'
- Recorra un vector de 20 valores de tipo word y cuente en **a0** cuántos valores son mayores que 400
- Dado un vector de 30 words, acumule en **a0** aquellos que están comprendidos entre 0x100 y 0x200
- Desarrollar un programa en ensamblador que dado un vector de enteros, obtenga como resultado cuántos elementos son iguales a cero. Este resultado se debe almacenar en una variable llamada *total*. El programa deberá inicializar los elementos del vector *V* en memoria, así como una variable *n* que almacenará el número de elementos que tiene el vector *V*.
- Desarrollar un programa en ensamblador que dado un vector de caracteres, cuente cuántas veces se repite un determinado caracter en el mismo. El programa deberá inicializar una cadena en memoria que finalice con el caracter nulo (un 0), una variable de tipo byte para colocar el caracter buscado y reservar un word para la variable que contendrá el resultado. Alinear la memoria después del byte. (analizar la directiva **.align**).