Estructura de Computadores

Grado de Ingeniería Informática

ETSINF

Ejemplos de programación en ensamblador

DISCA

Ejemplo 1

- Programa que calcula el perímetro de un rectángulo dados los valores de las longitudes de sus lados (25 y 30).

```
          .globl __start
          .text 0x00400000
__start:  li $t0,25
          li $t1,30
          add $s0,$t1,$t0
          add $s0,$s0,$s0
```

Ejemplo 2

- Programa que calcula el perímetro de un rectángulo dadas las longitudes de sus lados que están almacenadas en memoria de acuerdo con la siguiente declaración. El resultado también se dejará en memoria.

```
            .globl __start
            .data 0x10000000
A:          .word 25
B:          .word 30
P:          .space 4
```

Ejemplo 2

- Solución

```
          .text 0x00400000
__start:  la $t0,A
          la $t1,B
          la $t2,P
          lw $s0,0($t0)
          lw $s1,0($t1)
          add $s2,$s1,$s0
          add $s2,$s2,$s2
          sw $s2,0($t2)
```

Ejemplo 3

- Programa que recorre un vector de enteros e incrementa cada componente en una unidad

```
          .data 0x10000000
vector:   .word 3, -9, 2, 7
          .globl __start
          .text 0x00400000
```

Ejemplo 3

- Solución

  Recorremos el vector modificando directamente su puntero cada vez

```
__start:  la $s0, vector          # Puntero a vector[0]
          li $s1, 4               # Dimensión del vector
bucle:    lw $t0, 0($s0)          # Lee vector[i]
          addi $t0, $t0, 1        # Incrementa vector[i]
          sw $t0, 0($s0)          # Escribe vector[i]
          addi $s1, $s1, -1       # Decrementa contador
          addiu $s0, $s0, 4       # Actualiza puntero a vector[i+1]
          bgtz $s1, bucle
```

Ejemplo 4

- Programa que trasvasa los elementos del vector V1 al vector V2. La dimensión de los vectores se encuentra en la variable DIM

```
        .data 0x10001000
V1:     .word 3, -9, 2, 7

        .data 0x10002000
V2:     .space 16

        .data 0x10004000
DIM:    .word 4

        .globl __start
        .text 0x00400000
```

Ejemplo 4

- Solución

```
__start:  la $s0, V1          # Puntero a V1[0]
          la $s1, V2          # Puntero a V2[0]
          la $s2, DIM        # Puntero a dimensión del vector
          lw $t1, 0($s2)      # Lee dimensión del vector
bucle:    lw $t0, 0($s0)      # Lee V1[i]
          sw $t0, 0($s1)      # Escribe en V2[i]
          addi $t1, $t1, -1   # Decrementa contador
          addiu $s0, $s0, 4   # Actualiza puntero a V1[i+1]
          addiu $s1, $s1, 4   # Actualiza puntero a V2[i+1]
          bgtz $t1, bucle
```

Ejemplo 5

- Modificar el programa anterior de forma que antes de transvasar un elemento del vector V1 al V2 compruebe si éste es >=0. En ese caso trasvasará el valor tal cual, en caso contrario (<0) escribirá un 0

Ejemplo 5

- Solución

```
__start:  la $s0, V1          # Puntero a V1[0]
          la $s1, V2          # Puntero a V2[0]
          la $s2, DIM        # Puntero a dimensión del vector
          lw $t1, 0($s2)      # Lee dimensión del vector
bucle:    lw $t0, 0($s0)      # Lee V1[i]
          bgtz $t0, trasvasa  # Comprueba si >= 0
          sw $zero, 0($s1)    # Si <0 escribe un 0
          j contador          # salta a actualizar el contador
trasvasa: sw $t0, 0($s1)      # Escribe en V2[i]
contador: addi $t1, $t1, -1   # Decrementa contador
          addiu $s0, $s0, 4   # Actualiza puntero a V1[i+1]
          addiu $s1, $s1, 4   # Actualiza puntero a V2[i+1]
          bgtz $t1, bucle
```

Llamadas al Sistema (System Calls)

Algunas funciones simples

Nombre	$v0	Descripción	Argumentos	Resultado
print_int	1	Imprime el valor de un entero	$a0 = entero a imprimir	—
read_int	5	Lee el valor de un entero	—	$v0 = entero leido
exit	10	Acaba el proceso	—	—
print_char	11	Imprime un carácter	$a0 = carácter a imprimir	—

```
li $v0, 5          # Parámetro para la función  read_int
syscall            # Llamada a la función
sw $v0, valor      # Almacenamiento el valor leído en la memoria
```

Ejemplo 2 con visualización del perímetro

```
          .text 0x00400000
__start:  la $t0,A
          la $t1,B
          la $t2,P
          lw $s0,0($t0)
          lw $s1,0($t1)
          add $s2,$s1,$s0
          add $s2,$s2,$s2
          sw $s2,0($t2)
```

AÑADIRÍAMOS

```
move $a0,$s2    # copia el perímetro en $a0
li $v0,1        # código de print_int
syscall         # llamada al sistema
```

Ejemplo 6

- Implementar un programa que escriba los números del 1 al 10 y termine

Ejemplo 6

- Implementar un programa que escriba los números del 1 al 10 y termine

```
                .text 004000000
main:           addi $t1, $0, 10
                addi $a0, $0, 1
                li $v0, 1
                syscall
bucle:          addi $a0, $a0, 1
                li $v0, 1
                syscall
                bne $t1, $a0, bucle
                li $v0, 10
                syscall
```