UMU

Universidad de Murcia

Arturo Córdoba Pérez
Pablo José Rocamora Zamora

Prácticas 2 y 3

Práctica 2: ***Interger_to_string***

Parte 1

Nos ha resultado la parte más compleja de la práctica 2 por la necesidad de usar la operaciones ***lb*** y ***sb*** ya que en ese momento aún no comprendíamos bien su funcionamiento.

```
lb    $t3, 0($a2)
lb    $t4, 0($t0)
sb    $t3, 0($t0)
sb    $t4, 0($a2)
```

Parte 2

Lo que más nos ha costado en esta parte ha sido darnos cuenta de que para añadir el ***\0*** al final teníamos que sumar 1 en vez de sumar 4, ya que se trata de un ***byte*** y no un ***integer***.

```
addiu    $t0, $t0, 1
sb    $zero, 0($t0)          # *p = '\0'
```

Parte 3

Esta parte no nos ha supuesto ningun problema ya que solamente fue poner un ***if*** al principio para comprobar si el numero era ***0***.

```
abs    $t1, $a0        #move    $t1, $a0        # int i = n
bnez    $t1, B3_3
...

B3_3:
    blez    $t1, B3_6        # si i <= 0 salta el bucle
```

Parte 4

Implementar esta parte ha sido trivial partiendo de la anterior.

Fin de la práctica 2.

Práctica 3: ***Compara_enteros***

compara_enteros:

Esta función ha sido trivial de implementar, lo hemos hecho en el primer intento.

compara_vector_con_escalar:

Algunas de las erratas que hemos cometido en esta función y que más tarde averigüamos son:

1. Apilar y desapilar ***$a0*** porque pensabamos que era necesario.
2. En el ***for*** teníamos puesto ***bgt*** en vez de ***bge CS_for: bge $s0, $s6, CS_fin***.
3. Las 3 operaciones de suma las haciamos al principio, nos confundiamos por el ***++i*** del bucle ***for.***

```
addi    $s2, $s2, 1
addi    $s0, $s0, 1    #++i
addi    $s1, $s1, 4    #entero + 4
```

inicializa_vector:

El principal problema que teniamos era que cuando se inicializaba el vector con un tamaño inferior al anterior cuando hacia las comparaciones nos cogía un número de más del otro vector, por lo que al hacer la comparación estaba con un operador de más, estuvimos repasando todos los comparadores para ver si encontrabamos algún error pero no lo encontramos, así que lo que hicimos fue que en la función, antes de llenarla con los nuevos números, poner todo el vector anterior con ***\0***.

```
    la    $t1, cadena_resultado
    li    $t2, 0
IV_cle:    bge    $t2, $t0, IV_fisi    # blucle para vaciar cadena_resultado
    sw    $zero, 0($t1)
    addi    $t1, $t1, 4
    addi    $t2, $t2, 1
    j    IV_cle

IV_fisi: ...
```

Fin de la práctica 3.