



Instituto Politécnico Superior “General San Martín”

Tecnicatura en Informática Profesional y Personal

Instalación y Reemplazo de Comp. Internos

PRACTICA 1 MIPS

Estudiante: Lucas Wasserstrom

Cuestiones parte 1:

1. El valor que se carga en la posición res es de 1
2. Ahora se carga 0
3. La comparación que se evalúa es de (dato1 < dato2)
4. Hay que cambiar la instrucción slt por la instrucción seq.
- 5.

```
.data
dato1: .word 30
dato2: .word 30
res: .space 1
.text
main: lw $t0,dato1($0) # cargar dato1 en t0
      lw $t1,dato2($0) # cargar dato2 en t1
      sge $t2,$t0, $t1 # poner a 1 $t2 si t0>=t1
      sle $t3,$t0, $t1 # poner a 1 $t3 si t0<=t1
      and $t2, $t2, $t3 # poner a 1 $t2 si t2 == 1 && t3 == 1
      sb $t2,res($0) # almacenar $t2 en res
```

6. Se almacena 1 en la posición res
7. Ahora da 0 en res
8. El resultado de ahora es 1
9. Realiza una comparación de (dato1 <= dato2)
- 10.

```
.data
dato1: .word 40
dato2: .word 40
res: .space 1
.text
main: lw $t0,dato1($0) # cargar dato1 en t0
      lw $t1, dato2($0) # cargar dato2 en t1
      sle $t2, $t0, $t1 # poner a 1 $t2 si t0<=t1
      sb $t2,res($0) # almacenar $t2 en res
```

- 11.

```
.data
dato1: .word 30
dato2: .word 40
res: .space 1
.text
main: lw $t0,dato1($0) # cargar dato1 en t0
      lw $t1, dato2($0) # cargar dato2 en t1
      sgt $t2, $t0, $t1 # poner a 1 $t2 si t0>t1
      bne $t0,$t1,fineval # si t0<>t1 salta a fineval
      ori $t2,$0,1 # poner a 1 t3 si t0=t1
fineval: sb $t2,res($0) # almacenar $t2 en res
```

12.

```
.data
dato1: .word 40
dato2: .word 40
res: .space 1
.text
main: lw $t0,dato1($0) # cargar dato1 en t0
      lw $t1, dato2($0) # cargar dato2 en t1
      sge $t2, $t0, $t1 # poner a 1 $t2 si t0<=t1
      sb $t2,res($0) # almacenar $t2 en res
```

13. Se almacena 1 en res

14. Ahora se carga un 0

15. Aca tambien se carga un 0

16. En este también se almacena 0 en res

17. La comparación que se lleva a cabo es (dato1 <= 0 && dato2 <= 0)

18.

```
.data
dato1: .word 40
dato2: .word -50
res: .space 1
.text
main: lw $t8,dato1($0)
      lw $t9,dato2($0)
      and $t0,$t0,$0
      and $t1,$t1,$0
      beq $t8,$0,igual
      ori $t0,$0,1
igual: beq $t9,$t8,fineval
      ori $t1,$0,1
fineval: and $t0,$t0,$t1
      sb $t0,res($0)
```

19. Se almacena 1

20. Aquí se almacena 0 en res

21. Ahora se almacena 0 también

22. La comparación que se ha evaluado es (dato1 <= 0 && dato2 < dato1)

23.

```
.data
dato1: .word 30
dato2: .word -50
res: .space 1
.text
main: lw $t8,dato1($0)
      lw $t9,dato2($0)
      and $t1,$t1,$0
```

```

and $t0,$t0,$0
bne $t8,$t9,igual
ori $t0,$0,1
igual: sle $t1,$t8,$t9
fineval: and $t0,$t0,$t1
sb $t0,res($0)

```

24.

En el anterior ya lo había hecho con pseudoinstrucciones así que queda igual.

```

.data
dato1: .word 30
dato2: .word -50
res: .space 1
.text
main: lw $t8,dato1($0)
lw $t9,dato2($0)
and $t1,$t1,$0
and $t0,$t0,$0
bne $t8,$t9,igual
ori $t0,$0,1
igual: sle $t1,$t8,$t9
fineval: and $t0,$t0,$t1
sb $t0,res($0)

```

25. Se almacena 0 en res

26. Ahora almacena 1 el programa

27. En este también almacena 1 el programa

28. Aca se almacena 0 en res

29. La comparación que se realiza es (dato1 < dato2 || dato2 == 0)

30.

```

.data
dato1: .word 30
dato2: .word -20
res: .space 1
.text
main: lw $t8,dato1($0)
lw $t9,dato2($0)
and $t0,$t0,$0
and $t1,$t1,$0
sle $t0,$t8,$t9
ble $t9,$0,fineval
ori $t1,$0,1
fineval: or $t0,$t0,$t1
sb $t0,res($0)

```

31.

```

.data
dato1: .word 30
dato2: .word -20
res: .space 1
.text
main: lw $t8, dato1($0)
      lw $t9, dato2($0)
      and $t0, $t0, $0
      and $t1, $t1, $0
      sle $t0, $t8, $t9
      sle $t1, $t9, $0
      fineval: or $t0, $t0, $t1
      sb $t0, res($0)

```

Ejercicios parte 1:

Ejercicio 1:

```

.data
V: .byte 0, 1, 1, 1, 0
res: .space 3
.text

lb $s0, V
lb $s1, V+1
lb $s2, V+2
lb $s3, V+3
lb $s4, V+4

and $t0, $s0, $s4

sb $t0, res

or $t1, $s1, $s3

sb $t1, res+1

and $t0, $s1, $s2
or $t1, $t0, $s0

sb $t1, res+2

```

Ejercicio 2:

```

.data
V: .word 2, -4, -6
res: .space 12
.text

lw $s0, V
lw $s1, V+4
lw $s2, V+8

sge $t0, $s0, $0
sge $t1, $s1, $0
sge $t2, $s2, $0

sw $t0, res
sw $t1, res+4
sw $t2, res+8

```

Ejercicio 3:

```

.data
V: .word 1, -4, -5, 2
res: .space 1
.text

lw $s0, V
lw $s1, V+4
lw $s2, V+8
lw $s3, V+12

sge $t0, $s0, $0
sge $t1, $s1, $0
sge $t2, $s2, $0
sge $t3, $s3, $0

and $s4, $t0, $t1
and $s5, $t2, $t3
and $t4, $s4, $s5

sw $t4, res($0)

```

Cuestiones parte 2:

1. La instrucción que evalúa la condición es beq.

2. El beq sería el if, el then sería lo que se encuentra en la etiqueta “entonces” la cual sólo se ejecuta si beq no se cumple.
3. Se almacena un 71 en res.
4. Si dato1 es 0 se almacena 30 y si dato2 es 0 se almacena 40.
- 5.

```
.data
dato1: .word 40
dato2: .word 0
res: .space 4
.text
main: lw $t0,dato1($0) #cargar dato1 en $t0
lw $t1,dato2($0) #cargar dato2 en $t1
and $t2,$t2,$0 #t2=0
Si: ble $t1,$0,finsi #si $t1 = 0 finsi
entonces: div $t0,$t1 #t0/$t1
mflo $t2 #almacenar LO en $t2
finsi: add $t3,$t0,$t1 #$t3=$t0+$t1
add $t2,$t3,$t2 #$t2=$t3+$t2
sw $t2,res($0) #almacenar en res $t2
```

6. VARIABLES ENTERO: dato1=40; dato2=30; res;
INICIO
if (dato1!=0 && dato2!=0) res=dato1/dato2;
res=res+dato1+dato2;
FIN
7. Las condiciones que evalúan la condición del programa son los 2 beq que se hacen en el programa, los cuales serían la condición contraria del la condición del if en el pseudocódigo.
8. La estructura condicional del if está formada por 2 beq que forman la condición y el then esta formada por lo que sigue en la etiqueta “entonces”
9. Se almacena 71
10. Con dato1 = 0 ahora se almacena 30
Con dato2 = 0 ahora se almacena 40
- 11.

```
.data
dato1: .word 40
dato2: .word 0
res: .space 4
.text
main: lw $t0,dato1($0) #cargar dato1 en t0
lw $t1,dato2($0) #cargar dato2 en $t1
and $t2,$t2,$0 #pone a 0 $t2
Si: blt $t1,$0,finsi #si $t1=0 saltar finsi
ble $t0,$0,finsi #si $t0 =0 saltar finsi
entonces: div $t0,$t1 #t0/$t1
mflo $t2 #almacenar LO en t2
finsi: add $t3,$t0,$t1 #t3=t0+$t1
```

```
add $t2,$t3,$t2 #$t2=t3+$t2
sw $t2,res($0) #almacenar en res $t2
```

12. VARIABLES

ENTERO: dato1=30; dato2=40; res;

INICIO

Si ((dato1<=dato2)) entonces

res= dato1

Sino

res=dato2

FinSi

FIN

13. Se almacena 30 en res, y si dato1 == 35 entonces tambien se almacena 30 en res

14. El conjunto de instrucciones que implementan la pseudoinstrucción bge son slt y beq

15.

```
.data
dato1: .word 30
dato2: .word 40
res: .space 4
.text
main: lw $t0,dato1($0) #cargar dato1 en $t0
lw $t1,dato2($0) #cargar dato2 en $t1
Si: blt $t0,$t1, sino #si $t0>=$t1 ir a sino
entonces: sub $t0, $t0, $t1
sw $t0,res($0) #almacenar $t0 en res
j finisi #ir a finisi
sino: sub $t0, $t1, $t0
sw $t1,res($0)#almacenar $t1 en res
finisi:
```

16.

VARIABLES

ENTERO: dato1=30; dato2=40; dato3=-1; res;

INICIO

Si ((dato3>dato2) OR (dato3<dato1))

res = 1;

Sino

res=0;

FinSi

FIN

17. Para ambos casos el valor que se almacena en res 1.

18.

```
.data
dato1: .word 30
dato2: .word 40
dato3: .word -1
res: .space 4
```



```

.text
main: lw $t1,dato1($0) #cargar dato1 en $t1
      lw $t2,dato2($0) #cargar dato2 en $t2
      lw $t3,dato3($0) #cargar dato3 en $t3
      Si: blt $t3,$t1, sino #si $t3<$t1 ir sino
      bgt $t3,$t2, sino #si $t3<=$t2 ir a sino
      entonces: addi $t4,$0,1 #$t4=1
      j finisi #ir a finisi
      sino: and $t4,$0,$0 #$t4=0
      finisi: sw $t4,res($0) #almacenar res

```

19. Primero se cargan los datos, un vector que seria practicamente un string que dice "hola" y luego se toma espacio para una variable n, se carga en un registro la dirección de la cadena y se limpia el registro t2 para usarlo como contador. Luego se va cargando byte a byte las distintas letras de la palabra con lb y la dirección t0 que va aumentando en cada iteración para ir pasando de letra y siempre y cuando la letra no sea nula, osea 0, se le suma 1 a t2. Cuando termina el bucle gracias a que beq se fija si \$t1 == 0, se guarda en la dirección n el resultado \$t2 que es la cantidad de letras de la palabra.
20. Se almacena 4.
- 21.

```

.data
tira1: .asciiz "hola"
tira2: .asciiz "adios"
.align 2
n: .space 4
.text
main: la $t0,tira1 #carga dir. tira1 en $t0
      la $t1, tira2 #carga dir. tira2 en $t1
      andi $t3,$t3, 0 #$t2=0
      mientras: lb $s0,0($t0) #almacenar byte en $s0
                lb $s1, 0($t1) #almacenar byte en $s1

      beq $s0,$0,finmientras #si $s0=0 saltar a finmientras
      beq $s1, $0, finmientras #si $s1=0 saltar a finmientras

      addi $t2,$t2, 1 #$t2=$t2+1
      addi $t0,$t0, 1 #$t0=$t0+1
      addi $t1, $t1, 1 #$t1 = $t1+1
      j mientras #saltar a mientras
      finmientras: sw $t2,n($0) #almacenar $t2 en n

```

22. Se carga un vector de enteros [6,7,8,9,10,1] y se guarda espacio para la respuesta. Luego en el main se guarda en \$t2 la dirección del vector y se limpia \$t3, luego en \$t0 se carga un 0 y en \$t1 se carga un 6. Luego se lleva a cabo el for, el cual va a seguir siempre y cuando \$t0 < \$t1 osea mientras que \$t0 < 6, siendo este el contador del for. Después se carga en \$t4 con la dirección del vector un elemento de este mismo, ya que mientras pasan las iteraciones se va aumentando t2 para ir

corriendo por todos los elementos del vector. Se van sumando en \$t3, que habíamos limpiado anteriormente, los distintos elementos del array y se aumenta el contador \$t0 en 1. Cuando termina el for ya que la pseudoinstrucción bgt nos hace saltar a finpara, se guarda en la dirección de res el resultado de sumar todos los elementos del array

23. Se almacena 41.

24.

```
.data
v1: .word 6,7,8,9,10,-1, 34, 23
v2: .space 32
.text
main: la $t2,v1 #$t2=dirección de vector
la $s0, v2

and $t3,$0,$t3 #$t3=0

li $t0,0 #$t0=0
li $t1,8 #$t1=5
para: bgt $t0,$t1,finpara #si $t0>$t1 saltar finpara

lw $t4,0($t2) #carga elemento vector en $t4
addi $t4,$t4, 1 #suma 1 al elemento del vector

sw $t4, 0($s0) #guarda $t4 en el vector v2
addi $t2,$t2, 4 #$t2=$t2 + 4
addi $s0, $s0, 4 #$s0 = $s0 + 4
addi $t0,$t0, 1 #$t0=$t0 + 1
j para #saltar a bucle
finpara:
```