

Bài thực hành 3: Lệnh nhảy, rẽ nhánh

Họ tên: Nguyễn Văn Hiếu- 20225717

Assignment 1:

Gán $i=3$; $j=5$;

```
mips1.asm
1  .text
2  li $s1, 3
3  li $s2, 5
4  start:
5      slt $t0, $s2, $s1
6      bne $t0, $zero, else
7      addi $t1, $t1, 1
8      addi $t3, $zero, 1
9      j endif
10 else: addi $t2, $t2, -1
11      add $t3, $t3, $t3
12      endif:
13
```

Do giá trị $i < j$ nên thanh ghi \$t0 sau dòng lệnh slt có giá trị bằng 0.

⇒ Lệnh **bne** được thực hiện, do vậy hai lệnh 7 và 8 được thực hiện: giá trị thanh \$t1 và \$t2 tăng lên 1.

⇒ Lệnh thực hiện đúng.

Gán $i=5$, $j=3$;

```
1  .text
2  li $s1, 5
3  li $s2, 3
4  start:
5      slt $t0, $s2, $s1
6      bne $t0, $zero, else
7      addi $t1, $t1, 1
8      addi $t3, $zero, 1
9      j endif
10 else: addi $t2, $t2, -1
11      add $t3, $t3, $t3
12      endif:
13
```

Do giá trị $i > j$ nên thanh ghi \$t0 được gán bằng 1.

⇒ Lệnh **bne** được thực hiện rồi chuyển sang lệnh **else**. Sau đó thực hiện 2 dòng lệnh gán $y=y-1$; $z=2*z$.

⇒ Lệnh thực hiện đúng.

Assignment 2:

```
1
2 .data
3     A: .word 10,20,30,40,50
4 .text
5 addi $s5, $zero, 0 # sum = 0
6 addi $s1, $zero, 0 # i = 0
7 la $s2, A
8 li $s3, 5
9 li $s4, 1
10 loop: slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
11 beq $t2, $zero, endloop
12 add $t1, $s1, $s1 # $t1 = 2 * $s1
13 add $t1, $t1, $t1 # $t1 = 4 * $s1
14 add $t1, $t1, $s2 # $t1 store the address of A[i]
15 lw $t0, 0($t1) # load value of A[i] in $t0
16 add $s5, $s5, $t0 # sum = sum + A[i]
17 add $s1, $s1, $s4 # i = i + step
18 j loop # goto loop
19 endloop:
```

Gán giá trị ban đầu của mảng A lần lượt là 10, 20, 30, 40, 50 và gán i=0, sum=0, step=1.

Tổng (sum) được lưu ở thanh \$s5.

Ở mỗi vòng lặp dùng lệnh **lw** để lưu giá trị A[i].

Sau khi thực hiện hết các câu lệnh kết quả thu được ở thanh \$s5 là 0x00000096 đổi ra hệ thập phân là 150.

\$s5	21 0x00000096
------	---------------

⇒ Kết quả thu được đúng với lí thuyết.

Assignment 3:

Ban đầu gán giá trị test =1:

Lệnh **lw** thực hiện gán trị của thanh ghi \$s0 vào thanh ghi \$s1.

Sau đó lệnh beq được thực hiện

Do thanh ghi \$s1 = 1 có giá trị bằng với thanh ghi \$t1 nên dòng lệnh được nhảy đến **case_1** thực hiện câu lệnh tính $a=a-1$. Kết quả thu được trên thanh \$s2 là 0xffffffff đổi ra hệ thập phân là -1.

\$s2	18	0xffffffff
------	----	------------

⇒ Kết quả thu được đúng lý thuyết.

Gán giá trị test = 0:

Tương tự như trên thì dòng lệnh được nhảy đến **case_2** thực hiện câu lệnh tính $a=a+1$. Kết quả thu được trên thanh \$s2 là 0x00000001 đổi ra hệ thập phân là 1.

\$s2	18	0x00000001
------	----	------------

⇒ Kết quả thu được đúng với lý thuyết.

Gán giá trị test = 2:

Dòng lệnh được nhảy đến **case_3** thực hiện câu lệnh tính $b=2*b$. Kết quả thu được lưu ở thanh ghi \$s3 là 0x00000000.

\$s3	19	0x00000000
------	----	------------

⇒ Kết quả thu được đúng với lý thuyết.

Assignment 4:

a. $i < j$

#Laboratory Exercise 3, Home Assignment 1

.text

li \$s1, 3

li \$s2, 5

li \$t5, 1

start:

slt \$t0,\$s1,\$s2

bne \$t0,\$t5,else # branch to else if $j < i$

addi \$t1,\$t1,1 # then part: x=x+1

addi \$t3,\$zero,1 # z=1

j endif # skip “else” part

else: addi \$t2,\$t2,-1 # begin else part: y=y-1

add \$t3,\$t3,\$t3 # z=2*z

endif:

Kết quả thu được: (với i=3,j=5)

\$t1		9	0x00000001
\$t3		11	0x00000001

Kết quả thu được: (với i=5,j=3)

\$t2		10	0xffffffff
\$t3		11	0x00000000

b. $i \geq j$

#Laboratory Exercise 3, Home Assignment 1

.text

li \$s1, 5

li \$s2, 3

start:

slt \$t0,\$s1,\$s2

bne \$t0,\$zero,else # branch to else if j<i

addi \$t1,\$t1,1 # then part: x=x+1

addi \$t3,\$zero,1 # z=1

j endif # skip “else” part

else: addi \$t2,\$t2,-1 # begin else part: y=y-1

add \$t3,\$t3,\$t3 # z=2*z

endif:

Kết quả thu được: (i=5;j=3)

\$t1	9	0x00000001
\$t3	11	0x00000001

Kết quả thu được: (i=5;j=3)

\$t2	10	0xffffffff
\$t3	11	0x00000000

c. $i+j \leq 0$

#Laboratory Exercise 3, Home Assignment 1

.text

li \$s1, 3

li \$s2, 5

start:

add \$s1,\$s1,\$s2

slt \$t0,\$zero,\$s1

bne \$t0,\$zero,else # branch to else if j<i

addi \$t1,\$t1,1 # then part: x=x+1

addi \$t3,\$zero,1 # z=1

j endif # skip "else" part

else: addi \$t2,\$t2,-1 # begin else part: y=y-1

add \$t3,\$t3,\$t3 # z=2*z

endif:

Kết quả thu được: (i=3;j=5)

\$t2	10	0xffffffff
------	----	------------

\$t3	11	0x00000000
------	----	------------

Kết quả thu được: (i=4;i=-5)

\$t1	9	0x00000001
------	---	------------

\$t3	11	0x00000001
------	----	------------

d. $i+j > m+n$

#Laboratory Exercise 3, Home Assignment 1

.text

li \$s1, 4

li \$s2, 5

li \$s4, 3

li \$s5, 4

start:

add \$s1,\$s1,\$s2

add \$s4,\$s4,\$s5

slt \$t0,\$s1,\$s4

bne \$t0,\$zero,else # branch to else if $j < i$

addi \$t1,\$t1,1 # then part: $x = x + 1$

addi \$t3,\$zero,1 # $z = 1$

j endif # skip “else” part

else: addi \$t2,\$t2,-1 # begin else part: $y = y - 1$

add \$t3,\$t3,\$t3 # $z = 2 * z$

endif:

Kết quả thu được: (với $i=4, j=5, m=3, n=4$)

\$t1	9	0x00000001
\$t2	10	0x00000000
\$t3	11	0x00000001

Kết quả thu được với (i=4,j=5,m=3,n=10)

\$t2	10	0xffffffff
\$t3	11	0x00000000

Assignment 5:

a. $i \leq n$

#Laboratory 3, Home Assignment

.data

arr: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 # khởi tạo các phần tử của mảng arr

.text

addi \$s3, \$zero, 10 # số các phần tử của mảng arr

addi \$s4, \$zero, 1 # bước nhảy bằng 1

la \$s2, arr # lưu địa chỉ đầu tiên của mảng vào \$s2

addi \$s5, \$zero, 0 # sum = 0

addi \$s1, \$zero, 0 # i = 0

loop:

slt \$t2, \$s3, \$s1 # if $i \leq n$, goto endloop

bne \$t2, \$zero, endloop

add \$t1, \$s1, \$s1 # $t1 = 2 * s1$

add \$t1, \$t1, \$t1 # $t1 = 4 * s1$

add \$t1, \$t1, \$s2 # \$t1 store the address of A[i]

lw \$t0, 0(\$t1) # load value of A[i] in \$t0

add \$s5, \$s5, \$t0 # sum = sum + A[i]

add \$s1, \$s1, \$s4 # $i = i + \text{step}$

j loop # goto loop

endloop:

⇒ Vẫn thực hiện tính tổng các phần tử trong mảng.

Kết quả thu được:

\$s5	21	66
------	----	----

b. $\text{sum} \geq 0$

#Laboratory 3, Home Assignment

.data

arr: .word 1, 2, 3, 4, 5, -20, 6, 7, 8, 9, 10, 11 # khởi tạo các phần tử của mảng arr

.text

```

addi $s3, $zero, 10 # số các phần tử của mảng arr
addi $s4, $zero, 1 # bước nhảy bằng 1
la $s2, arr # lưu địa chỉ đầu tiên của mảng vào $s2
addi $s5, $zero, 0 # sum = 0
addi $s1, $zero, 0 # i = 0
loop:
    slt $t2, $s5, $zero # if i<=n, goto endloop
    bne $t2, $zero, endloop
add $t1, $s1, $s1 # $t1 = 2 * $s1
add $t1, $t1, $t1 # $t1 = 4 * $s1
    add $t1, $t1, $s2 # $t1 store the address of A[i]
lw $t0, 0($t1) # load value of A[i] in $t0
add $s5, $s5, $t0 # sum = sum + A[i]
add $s1, $s1, $s4 # i = i + step
    j loop # goto loop
endloop:
    Khi sum<0 kết thúc vòng lặp.
    Kết quả thu được:

```

\$s5	21	-5
------	----	----

c. $A[i]! = 0$

#Laboratory 3, Home Assignment

.data

arr: .word 1, 2, 3, 4, 5, 0, 6, 7, 8, 9, 10, 11 # khởi tạo các phần tử của mảng arr

.text

```

addi $s3, $zero, 10 # số các phần tử của mảng arr
addi $s4, $zero, 1 # bước nhảy bằng 1
la $s2, arr # lưu địa chỉ đầu tiên của mảng vào $s2
addi $s5, $zero, 0 # sum = 0
addi $s1, $zero, 0 # i = 0
loop:
    #slt $t2, $s5, $zero # if i<=n, goto endloop

add $t1, $s1, $s1 # $t1 = 2 * $s1
add $t1, $t1, $t1 # $t1 = 4 * $s1
add $t1, $t1, $s2 # $t1 store the address of A[i]
lw $t0, 0($t1) # load value of A[i] in $t0
beq $t0, $zero, endloop
add $s5, $s5, $t0 # sum = sum + A[i]
add $s1, $s1, $s4 # i = i + step

```


j loop # goto loop

endloop:

Khi A[i]=0 kết thúc vòng lặp.

Kết quả:

§ 5	21	15
-----	----	----

Assignment 6:

.data

A: .word -13, 7, 12, -5, 8, 2, 10, -3, 6, 4

n: .word 10

.text

.globl main

main:

la \$s0, A #Luu dia chi dau tien cua mang vao \$s0

lw \$t0, n(\$zero) #Load so luong phan tu vao \$t0

li \$s1, 0 #Dat gia tri tuyet doi lon nhat ban dau bang 0

j loop

loop:

beq \$t0, \$zero, exit #Neu tat ca cac phan tu da duockiem tra, thoat vong lap

lw \$t1, 0(\$s0) #Load gia tri hien tai cuar phan tu vao \$t1

abs \$t2, \$t1 #Tinh gia tri tuyet doi cua phan tu va luu vao \$t2

bge \$t2, \$s1, find_max #Neu gia tri tuyet doi lon hon gia tri tuyet doi hien tai, cap nhat lai GTLN

addi \$s0, \$s0, 4 #Tang dia chi de xet phan tu tiep theo

addi \$t3, \$zero, -1 #Luu tru gia tri -1 vao \$t3

add \$t0, \$t0, \$t3 # Giam so luong ptu can tim kiem di 1

j loop

find_max:

move \$s1, \$t2 #Cap nhat gia tri tuyet doi lon nhat

addi \$s0, \$s0, 4 #Tang dia chi de xet ptu tiep theo

addi \$t3, \$zero, -1 #Luu tru gia tri -1 vao \$t3

add \$t0,\$t0,\$t3 #Giam so luong ptu tim kiem di 1

exit:

li \$v0, 1 #Dat ma lenh chp syscall de in ket qua

move \$a0, \$s1 #Load gia tri tuyet doi lon nhat vao \$a0

syscall

li \$v0, 10 #Dat ma lenh cho syscall de ket thuc chuong trinh

syscall #Thoat chuong trinh

Kết quả thu được:

