

LAB#2

INSTRUCTOR: SAM, X. NGUYEN

STUDENT ID: 20110243

| FULL IN NAME: Lê Hải Đăng

Conditional branch (BNE)-page#90

.data

.text

Giả sử chúng ta có 4 biến: g,h,i,j tương ứng với các giá trị 4,3,2,1. Chúng ta lần lượt gán nó vào thanh ghi s1, s2, s3, s4. (li: load input, dùng để đưa giá trị vào thanh ghi).

li \$s1,4 *# g=4*

li \$s2,3 *# h=3*

li \$s3,2 *# i=2*

li \$s4,1 *# j=1*

bne \$s3, \$s4, Else *# bne (branch not equal). Nếu giá trị tại s3 không bằng s4 thì chương trình sẽ nhảy đến địa chỉ "Else", ngược lại thì chương trình tiếp tục thực hiện.*

add \$s0, \$s1, \$s2 *# add (cộng). Bước này sẽ thực hiện cộng giá trị tại s1,s2 và lưu vào s0 (s0 = s1 + s2) <=> (f = g + h).*

In kết quả

li \$v0,1 *# Mã 1 dùng để in số nguyên.*

move \$a0, \$s0 *# Di chuyển s0 vào a0.*

syscall *# Lệnh syscall làm treo sự thực thi của chương trình và chuyển quyền điều khiển cho HĐH (được giả lập bởi MARS).*

j Exit *# Được hiểu là chương trình sẽ nhảy đến nhãn Exit để thực thi tiếp tục.*

Else: sub \$s0, \$s1, \$s2 *# Nếu điều kiện phía trên (bne) không đúng sẽ tiếp tục thực hiện bước này; sub (trừ). Bước này sẽ thực hiện trừ giá trị tại s1,s2 và lưu vào s0. (s0 = s1 - s2) <=> (f = g - h).*

In kết quả

li \$v0, 1 *# Mã 1 dùng để in số nguyên.*

move \$a0, \$s0 *# Di chuyển s0 vào a0.*

syscall

Lệnh syscall làm treo sự thực thi của chương trình và chuyển quyền điều khiển cho HĐH (được giả lập bởi MARS).

Exit:

li \$v0,10

Mã 10 dùng để thoát chương trình.

Syscall

Lệnh syscall làm treo sự thực thi của chương trình và chuyển quyền điều khiển cho HĐH (được giả lập bởi MARS).

Kết quả thực thi:

