LAB#3

INSTRUCTOR: SAM, X. NGUYEN

STUDENT ID: 20110243 | FULL IN NAME: Lê Hải Đăng

##########################

# int fact (int n)

#{

#if (n < 1) return (1);

#else return (n \* fact(n – 1));

#}

.text

main:

li $a0, 6 *# 6!=6\*5\*4\*3\*2\*1*

jal fact *# jal (jump and link). Dòng này chương trình sẽ thực thi hàm fact.*

move $a0, $v0 *# di chuyển (sao chép) giá trị $v0 vào $a0.*

li $v0, 1 *# code = 0, print*

syscall

exit:

li $v0, 10 *# code =10, exit*

syscall

fact:

beq $a0, 1, L1 *# beq (branch equal quality). Nếu giá trị tại $a0 bằng 1 thì chương trình sẽ nhảy đến địa chỉ “L1”, ngược lại thì chương trình tiếp tục thực hiện.*

addi $sp, $sp, -8 *# Mở rộng stack để sử dụng (2 item)*

sw $ra, 0($sp) *# Lưu $ra vào stack.*

sw $a0, 4($sp) *# Lưu $a0 vào stack.*

addi $a0, $a0, -1 *# Giảm giá trị #a0 đi 1 đơn vị (a0 = a0 - 1).*

jal fact *# Gọi lại hàm fact với a0 đã giảm đi 1 đơn vị, $ra bị thay đổi*

lw $a0, 4($sp) *# Khôi phục $a0.*

lw $ra, 0($sp) *# Khôi phục $ra.*

addi $sp, $sp, 8 *# Khôi phục $sp.*

mul $v0, $a0, $v0 *# Thực hiện phép tính v0 = a0 \* v0*

j $ra *# Nhảy đến địa chỉ quay về $ra.*

L1:

li $v0, 1 *# Gán v0 = 1.*

j $ra *# Nhảy đến địa chỉ quay về $ra.*

########################################

# rewrite the program using

# call n from keyboard

# check n using the lines :

# slti $t0,$a0,1 # test for n < 1

# beq $t0,$zero,L1 # if n >= 1, go to L1.

########################################

.data

msg1: .asciiz "Nhap so muon tinh giai thua: "

msg2: .asciiz "Ban nhap so khong phu hop de tinh giai thua!!!\n"

.text

main:

INPUT:

li $v0,4 *# mã 4 dùng để in msg1*

la $a0, msg1

syscall

li $v0,5 *# mã 5 dùng để đọc (int) input*

syscall

move $a0,$v0

slti $t0,$a0,1 *# lệnh này sẽ gán $t0 bằng 1 nếu $a0<1, bằng 0 trong trường hợp ngược lại.*

bne $t0,$zero, L2 *# so sánh $t0 và $zero, nếu $t0 != 0 sẽ nhảy đến L2, ngược lại tiếp tục thực thi chương trình.*

jal fact *# jal (jump and link). Dòng này chương trình sẽ thực thi hàm fact.*

move $a0, $v0 *# di chuyển (sao chép) giá trị $v0 vào $a0.*

li $v0, 1 *# code = 0, print*

syscall

exit:

li $v0, 10 *# code =10, exit*

syscall

fact:

beq $a0, 1, L1 *# beq (branch equal quality). Nếu giá trị tại $a0 bằng 1 thì chương trình sẽ nhảy đến địa chỉ “L1”, ngược lại thì chương trình tiếp tục thực hiện.*

addi $sp, $sp, -8 *# Mở rộng stack để sử dụng (2 item)*

sw $ra, 0($sp) *# Lưu $ra vào stack.*

sw $a0, 4($sp) *# Lưu $a0 vào stack.*

addi $a0, $a0, -1 *# Giảm giá trị #a0 đi 1 đơn vị (a0 = a0 - 1).*

jal fact *# Gọi lại hàm fact với a0 đã giảm đi 1 đơn vị, $ra bị thay đổi*

lw $a0, 4($sp) *# Khôi phục $a0.*

lw $ra, 0($sp) *# Khôi phục $ra.*

addi $sp, $sp, 8 *# Khôi phục $sp.*

mul $v0, $a0, $v0 *# Thực hiện phép tính v0 = a0 \* v0*

j $ra *# Nhảy đến địa chỉ quay về $ra.*

L1:

li $v0, 1 *# Gán v0 = 1.*

j $ra *# Nhảy đến địa chỉ quay về $ra.*

L2:

li $v0,4 *# mã 4 dùng để in msg2*

la $a0, msg2

syscall

j INPUT *# Nhảy đến nhãn input*

