**BÁO CÁO THỰC HÀNH**

**IT3280 – 156788 – THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**NỘI DUNG**

**Bài 7. Lệnh gọi chương trình con, truyền tham số sử dụng ngăn xếp**

|  |  |
| --- | --- |
| **Họ và tên** | Nguyễn Minh Quân |
| **Mã số sinh viên** | 20235816 |

**Assignment 1**

**Tạo project để thực hiện Home Assignment 1. Dịch và chạy mô phỏng. Thay đổi các tham số chương trình (thanh ghi a0) và quan sát kết quả thực hiện. Chạy chương trình ở chế độ từng dòng lệnh và chú ý sự thay đổi của các thanh ghi, đặc biệt là thanh ghi pc và ra.**

**Home Assignment 1**

Chương trình dưới đây minh họa việc khai báo và sử dụng hàm abs để tính giá trị tuyệt đối của một số nguyên. Hàm sử dụng 2 thanh ghi, a0 chứa tham số vào và s0 chứa kết quả. Đọc kỹ chương trình và hiểu cách khai báo và gọi chương trình con.

Chương trình thực hiện:

A screenshot of a computer program

AI-generated content may be incorrect.

**Chạy chương trình ở chế độ từng dòng lệnh và chú ý sự thay đổi của các thanh ghi, đặc biệt là thanh ghi pc và ra.**

**Lệnh 1: li a0, -20** #load -20 vào a0, a0 = -20







**Lệnh 2: jal abs nhảy đến label abs và lưu địa chỉ của lệnh tiếp theo (lệnh li a7, 10) vào thanh ghi ra**





**Lệnh 3:sub s0, zero, a0 # s0 = -a0 = 20**

****

****



**Lệnh 4: blt a0, zero, done # a0 = -20 < 0 🡪 Nhảy tới label done**

****

****

****

**Lệnh 5: jr ra # Nhảy tới địa chỉ lưu trong thanh ghi ra( địa chỉ của li a7,10)**





Kết quả:





**Assignment 2**

**Tạo project để thực hiện Home Assignment 2. Dịch và chạy mô phỏng. Thay đổi các tham số chương trình (thanh ghi a0, a1, a2) và quan sát kết quả thực hiện. Chạy chương trình ở chế độ từng dòng lệnh và chú ý sự thay đổi của các thanh ghi, đặc biệt là thanh ghi pc và ra.**

**Home Assignment 2**

Trong ví dụ này, chương trình con max được khai báo và sử dụng để tìm phần tử lớn nhất trong 3 số nguyên. Các tham số này được truyền vào chương trình con qua các thanh ghi a0, a1, a2. Kết quả được lưu vào thanh ghi s0. Đọc kỹ chương trình và hiểu cách khai báo và gọi chương trình con.

Chương trình thực hiện:

A screenshot of a computer program

AI-generated content may be incorrect.

**Chạy chương trình ở chế độ từng dòng lệnh và chú ý sự thay đổi của các thanh ghi, đặc biệt là thanh ghi pc và ra.**

**Lệnh 1:**

**li a0, 4 # a0 =4**

**li a1, 8 # a1 =8**

**li a2, 3 #a2=3**

**A green and white flag

AI-generated content may be incorrect.**

**Lệnh 2:**

**jal max # nhảy tới nhãn max, lưu địa chỉ trở về(câu lệnh tiếp theo)vào ra**

****

****

**Lệnh 3: add s0, a0, zero # max = s0 = a0**

**** ****

****

**Lệnh 4: sub t0, a1, s0 # So sánh xem a1 có lớn hơn s0 hiện tại hay ko, t0 = a1-s0**

****

****

****

**Lệnh 5:** **blt t0, zero, okay # Nếu a1 – s0 < 0 hay a1 < max thì nhảy tới nhãn okay kiểm tra tiếp**

****

****

**Lệnh 6:** **add s0, a1, zero # Còn không thì update max = s0 = a1**

****

****

****

**Lệnh 7: sub t0, a2, s0 # Tiếp tục check xem a2 có lớn hơn max hay không**

****

****

****

**Lệnh 8: blt t0, zero, done # Nếu a2 – s0 < 0 thì không update gì cả, nhảy tới nhãn done**

****

****

****

**Lệnh 9: jr ra # quay lại địa chỉ đã lưu trong thanh ra**

****

****

**Kết quả:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Assignment 3**

**Tạo project để thực hiện Home Assignment 3. Dịch và chạy mô phỏng. Thay đổi tham số chương trình (thanh ghi s0, s1), quan sát quá trình và kết quả thực hiện. Chú ý sự thay đổi giá trị của thanh ghi sp. Quan sát vùng nhớ được trỏ bởi thanh ghi sp trong cửa sổ Data Segment.**

**Home assigment3**

Chương trình hợp ngữ dưới đây minh họa cách sử dụng vùng nhớ ngăn xếp (stack) với hai phép toán push và pop bằng cách sử dụng lệnh lw và sw. Giá trị của hai thanh ghi s0 và s1 sẽ được hoán đổi với nhau sử dụng ngăn xếp.

Chương trình thực hiện:

A screenshot of a computer program

AI-generated content may be incorrect.

**Quan sát quá trình và kết quả thực hiện. Chú ý sự thay đổi giá trị của thanh ghi sp. Quan sát vùng nhớ được trỏ bởi thanh ghi sp trong cửa sổ Data Segment.**

Lệnh 1:

li s0,5 #s0 = 5

li s1,6 # s0 =6



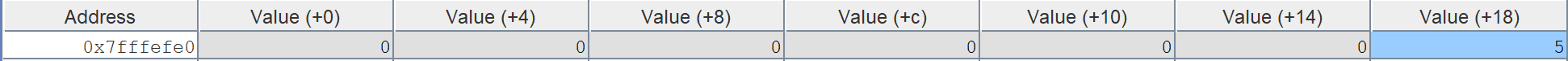
Lệnh 2:

addi sp, sp, -8 # sp = sp -8, tạo không gian 8 byte trên stack



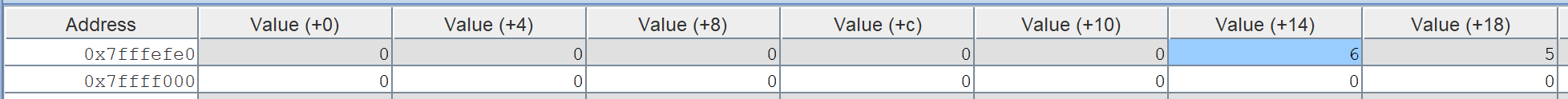
Lệnh 3:

sw s0, 4(sp) # Lưu s0=5 vào địa chỉ sp +4



Lệnh 4:

sw s1, 0(sp) # Lưu s1=6 vào địa chỉ sp + 0



Lệnh 5:

lw s0, 0(sp) # Tải giá trị từ sp + 0(6) vào thanh ghi s0 🡪 s0 =6



Lệnh 6:

lw s1, 4(sp) #Tải giá trị từ sp +4(5) vào thanh ghi s1 🡪 s1 = 5



Lệnh 7:

addi sp, sp, 8 # sp = sp +8, giải phóng bộ nhớ trên stack



Kết quả:



**Assignment 4**

**Tạo project để thực hiện Home Assignment 4. Dịch và chạy mô phỏng. Thay đổi tham số ở thanh ghi a0 và kiểm tra kết quả ở thanh ghi s0. Chạy chương trình ở chế độ từng dòng lệnh và quan sát sự thay đổi giá trị của các thanh ghi pc, ra, sp, a0, s0. Liệt kê các giá trị trong vùng nhớ ngăn xếp khi thực hiện chương trình với n = 3.**

**Chương trình thực hiện**

**# Laboratory Exercise 7, Home Assignment 4**

**.data**

**message: .asciz "Ket qua tinh giai thua la: "**

**.text**

**main:**

**jal WARP**

**print:**

**add a1, s0, zero # a0 = result from N!**

**li a7, 56**

**la a0, message**

**ecall**

**quit:**

**li a7, 10 # terminate**

**ecall**

**end\_main:**

**# ----------------------------------------------------------------------**

**# Procedure WARP: assign value and call FACT**

**# ----------------------------------------------------------------------**

**WARP:**

**addi sp, sp, -4 # adjust stack pointer**

**sw ra, 0(sp) # save return address**

**li a0, 3 # load test input N**

**jal FACT # call fact procedure**

**lw ra, 0(sp) # restore return address**

**addi sp, sp, 4 # return stack pointer**

**jr ra**

**wrap\_end:**

**# ----------------------------------------------------------------------**

**# Procedure FACT: compute N!**

**# param[in] a0 integer N**

**# return s0 the largest value**

**# ----------------------------------------------------------------------**

**FACT:**

**addi sp, sp, -8 # allocate space for ra, a0 in stack**

**sw ra, 4(sp) # save ra register**

**sw a0, 0(sp) # save a0 register**

**li t0, 2**

**bge a0, t0, recursive**

**li s0, 1 # return the result N!=1**

**j done**

**recursive:**

**addi a0, a0, -1 # adjust input argument**

**jal FACT # recursive call**

**lw s1, 0(sp) # load a0**

**mul s0, s0, s1**

**done:**

**lw ra, 4(sp) # restore ra register**

**lw a0, 0(sp) # restore a0 register**

**addi sp,sp,8 # restore stack pointer**

**jr ra # jump to caller**

**fact\_end:**

**Kết quả với n=3**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Giá trị trong vùng nhớ ngăn xếp khi thực hiện chương trình với n = 3:**

|  |
| --- |
| **bottom** |
| **0x00400004** |
| **0x00400030** |
| **3 (= a0)** |
| **0x00400060** |
| **2 (=a0)** |
| **0x00400060** |
| **1 (=a0)** |

**Assignment 5**

**Viết chương trình con tìm giá trị lớn nhất, nhỏ nhất và vị trí tương ứng trong danh sách gồm 8 số nguyên được lưu trữ trong các thanh ghi từ a0 đến a7. Ví dụ: Largest: 9, 3 => Giá trị lớn nhất là 9 được lưu trữ trong a3 Smallest: -3, 6 => Giá trị nhỏ nhất là -3 được lưu trữ trong a6 Gợi ý: Sử dụng ngăn xếp để truyền tham số.**

**Chương trình thực hiện:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Kết quả thu được:**

**A screenshot of a computer

AI-generated content may be incorrect.**