***Họ và tên: Nguyễn Đỗ Quang***

***MSSV: 20520720***

**BÀI THỰC HÀNH LAB 3**

***2 Thực hành:***

*2.1 Một số lệnh assembly cơ bản:*

|  |  |  |
| --- | --- | --- |
| Câu lệnh | Cú pháp | Chức năng |
| add | add $t0,$t1,$t2 | Cộng giá trị của thanh ghi $t1,$t2 và gán vào $t0 |
| addi | add $t0,$t1,1 | Cộng giá trị của thanh ghi $t1 và số nguyên rồi gán vào $t0 |
| addiu | addiu $t0,$t1,1 | Cộng giá trị của thanh ghi $t1 và số nguyên dương rồi gán vào $t0 |
| sub | sub $t0,$t1,$t2 | Trừ giá trị của thanh ghi $t1,$t2 và gán vào $t0 |
| subu | subu $t0,$t1,1 | Trừ giá trị của thanh ghi $t1 và số nguyên dương rồi gán vào $t0 |
| sw | sw $t0,($t1) | $t0 giá trị vào địa chỉ $t1 |
| slti | slti $t0,$t1,1 | trả về 1 nếu gía trị $t1<(số nguyên) |
| slt | slt $t0,$t1,$t2 | trả về 1 nếu gía trị $t1<$t2 |
| and | and $t0,$t1,$t2 | and từng bit giữa 2 giá trị |
| nor | nor $t0,$t1,$t2 | nor từng bit giữa 2 giá trị |
| or | or $t0,$t1,$t2 | or từng bit giữa 2 giá trị |
| syscall |  | lời gọi hệ thống |

*2.2 Mô phỏng các chương trình:*

***Ví dụ 1: Khai báo biến và gán giá trị của biến vào thanh ghi***

**Code:**

.data **# Phần khai báo biến**

var1: .word 23 **# Gán 23 vào biến var1**

.text **# Phần code**

\_\_start:

lw $t0, var1 **#load giá trị của biến var1 vào $t0**

li $t1, 5 **#load số 5 vào thanh ghi $t1**

sw $t1,var1 **#Gán giá trị 5 của thanh ghi $t1 vào biến var1**

***Ví dụ 2: Cấp phát mảng***

**Code:**

.data

array1: .space 12 ***#Cấp phát mảng 12 bytes***

.text

\_\_start:

la $t0, array1 **#load địa chỉ của array1 vào $t0**

li $t1, 5 **#load số 5 vào thanh ghi $t1**

sw $t1, ($t0) **#load giá trị của thanh ghi $t1 vào array[1]**

li $t1,13 **#load số 13 vào thanh ghi $t1**

sw $t1,4($t0) **#load giá trị của thanh ghi $t1 vào array[2]**

li $t1,-7 **#load số -7 vào thanh ghi $t1**

sw $t1,8($t0) **#load giá trị của thanh ghi $t1 vào array[3]**

***Ví dụ 3: Đọc input từ bàn phím***

**Code:**

li $v0, 5 **#Lệnh syscall 5 *để đọc số nguyên từ bàn phím***

syscall

***Ví dụ 4: In chuỗi ra màn hình***

**Code:**

.data

string1: .asciiz "Print this.\n" #***Khai báo biến string1***

.text

main:

li $v0,4 ***#Lệnh syscall 4 để xuất ra màn hình kết quả***

la $a0, string1 **#Load địa chỉ của string1 và $a0 để chuẩn bị xuất ra màn hình**

syscall

***3 Bài tập:***

***a Nhập vào một chuỗi, xuất ra cửa sổ I/O của MARS***

**Code**

.data # Phần khai báo biến

string1: .asciiz "Chao ban! Ban la sinh vien nam thu may?" #***Khai báo biến string1***

string2: .asciiz "Hihi, minh la sinh vien nam thu 1 ^-^" ***#Khai báo biến string2***

.text # Phần code

li $v0,4 ***#Load lệnh syscall 4 print screen***

la $a0,string1 ***#Load địa chỉ của biến string1 vào $a0 để chuẩn bị xuất ra***

syscall

la $a0,string2 ***#Load địa chỉ của biến string1 vào a0 để chuẩn bị xuất ra***

syscall

***b Biểu diễn nhị phân của 2 chuỗi trên dưới bộ nhớ***

Biểu diễn nhị phân của 2 chuỗi trên dưới bộ nhớ ở dưới dạng **little endian**

sử dụng 4 bytes, mỗi byte để biểu diễn một chữ cái, và được biểu diễn ngược

|  |  |  |
| --- | --- | --- |
| địa chỉ | value(+0) | value(+4) |
| giá trị | o a h C | n a b \_space\_ |
| hex | 6f 61 68 43 | 6e 61 62 20 |

***c Xuất ra lại đúng chuỗi đã nhập***

**Code**

.data

buffer: .space 20 ***#Cấp phát mảng 20 bytes***

.text

li $v0,8 ***#Lệnh syscall 8 để đọc chuỗi từ bàn phím***

li $a1,100 ***#Số kí tự cho phép đọc tối đa (100)***

la $a0,buffer ***#load địa chỉ của mảng vào $a0***

syscall

li $v0,4 ***#In dòng chữ vừa mới nhập ra màn hình***

syscall

***d Nhập vào 2 số nguyên sau đó xuất tổng của 2 số nguyên này***

**Code**

li $v0,5 ***#Lệnh 5 để đọc số nguyên***

syscall

addi $t0,$v0,0 ***#Ghi số vừa đọc vào thanh ghi $t0***

li $v0,5 ***#Lệnh 5 để đọc số nguyên***

syscall

addi $t1,$v0,0 ***#Ghi số vừa đọc vào thanh ghi $t0***

add $a0,$t0,$t1 ***#Cộng 2 số vừa đọc và ghi vào thanh ghi $a0***

li $v0,1 ***#Lệnh syscall 4 để xuất ra màn hình kết quả***

syscall

***Link video demo:*** <https://drive.google.com/file/d/10sLbfLY0e4qYe5G3rNm89oQpQ2ySVcXu/view?usp=sharing>