

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH



UIT
Trường Đại học
Công nghệ Thông tin

**Khoa Khoa học
và Kỹ thuật Thông tin**

Nguyễn Lê Quỳnh Hương – 21520255
IT012.N11.CNCL

BÁO CÁO:

Lab 3

GVHD: Nguyễn Văn Tín

Hồ Chí Minh, 2022

1. Lý thuyết

2. Thực hành :

2.1 Một số lệnh assembly MIPS cơ bản :

Lệnh	Mô phỏng	Chức năng
add	add \$1,\$2,\$3	Tính tổng hai giá trị trong thanh ghi \$2, \$3 rồi lưu vào thanh ghi \$1.
addi	addi \$1,\$2,200	Tính tổng giá trị trong thanh ghi \$2 với hằng số bất kỳ (200), lưu vào giá trị \$1.
addu	addu \$1,\$2,\$3	Tính tổng hai giá trị trong thanh ghi \$2, \$3 rồi lưu vào thanh ghi \$1 (giá trị sẽ được xem như số không dấu khi tổng bị tràn)
addiu	addiu \$1,\$2,200	Tính tổng giá trị trong thanh ghi \$2 với một số (200), lưu vào giá trị \$1 (giá trị sẽ được xem như số không dấu khi tổng bị tràn)
sub	sub \$1,\$2,\$3	Tính hiệu giá trị của thanh ghi \$2 với \$3 rồi lưu vào thanh ghi \$1
subu	subu \$1, \$2, 200	Tính hiệu giá trị của thanh ghi \$2 với một số (200), lưu vào giá trị \$1 (giá trị sẽ được xem như số không dấu khi tổng bị tràn)

and	and \$1,\$2,\$3	Thao tác bit AND hai giá trị của thanh ghi \$2 với \$3, lưu kết quả vào thanh ghi \$1
andi	andi \$1,\$2,200	Thao tác bit AND giá trị của thanh ghi \$2 với một số (200) rồi lưu kết quả vào thanh ghi \$1
or	or \$1,\$2,\$3	Thao tác bit OR hai giá trị của thanh ghi \$2 với \$3, lưu kết quả vào thanh ghi \$1
nor	nor \$1,\$2,\$3	Thao tác bit NOR hai giá trị của thanh ghi \$2 với \$3, lưu kết quả vào thanh ghi \$1
lw	lw \$1, 0(\$2)	Chuyển dữ liệu (word) từ bộ nhớ vào thanh ghi \$1
sw	sw \$1, 0(\$2)	Lệnh chuyển dữ liệu (word) từ thanh ghi \$1 ra bộ nhớ
slt	slt \$1,\$2,\$3	Kiểm tra xem \$2 có bé hơn \$3 không nếu đúng, \$1=1 ngược lại \$1=0
slti	slti \$1,\$2,200	Kiểm tra xem \$2 có bé hơn một số (200) không nếu đúng, \$1=1 ngược lại \$1=0
sltu	sltu \$1,\$2,\$3	Kiểm tra xem \$2 có bé hơn \$3 không nếu đúng, \$1=1 ngược lại \$1=0 (\$2, \$3 chứa số không dấu)

sltui	sltui \$1,\$2,200	Kiểm tra xem \$2 có bé hơn một số (200) không nếu đúng, \$1=1 ngược lại \$1=0 (\$2, \$3 chứa số không dấu)
syscall	syscall	Dùng để yêu cầu dịch vụ từ kernel, mã dịch vụ được truyền từ thanh ghi \$v0

2.2 Mô phỏng các chương trình bên dưới và có biết ý nghĩa của chương trình:

code	Giải thích
<pre>.data var1: .word 23 .text __start: lw \$t0, var1 li \$t1, 5 sw \$t1, var1</pre>	<p>Khai báo vùng nhớ data</p> <p>Khai báo biến kiểu word: var1 = 23</p> <p>Khai báo vùng nhớ text</p> <p>t0 lưu giá trị var1</p> <p>t1 = 5</p> <p>var1 lưu giá trị t1</p>

code	Giải thích
<pre>.data array1: .space 12 .text __start: la \$t0, array1 li \$t1, 5 sw \$t1, (\$t0) li \$t1, 13</pre>	<p>Cấp 12-byte bộ nhớ, chưa được khởi tạo</p> <p>t0 = địa chỉ array1</p> <p>t1 = 5</p> <p>array1[0] = t1</p> <p>t1 = 13</p>

sw \$t1, 4(\$t0)	array[1] = t1
li \$t1, -7	t1 = -7
sw \$t1, 8(\$t0)	array1[2] = t1

code	Giải thích
li \$v0,5	Truyền tham số 5 vào thanh ghi v0
syscall	Thực hiện chức năng (đọc số nguyên)

code	Giải thích
.data string1: .ascii "Print this.\n"	Khai báo mảng string1 = "Print this.\n"
.text main: li \$v0,4 la \$a0,string1 syscall	Truyền tham số 4 vào v0 (thực hiện chức năng in chuỗi kí tự mà địa chỉ được lưu trong a0 a0 = địa chỉ string1 Thực hiện chức năng

3. Bài tập :

3.1 Nhập vào một chuỗi, xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

a)

.data

string1: .ascii "Chao ban! ban la sinh vien nam may?"

string2: .ascii "\nHihi,minh la sinh vien nam 1 ^_^"

.text

```
main:      li $v0,4
           la $a0,string1
           syscalli $v0,4
           la $a0,string2
           syscall
```

b) Biểu diễn nhị phân của 2 chuỗi trên dưới bộ nhớ là gì?

0000000010010100001010000000100000

c)

data

string1: .asciiz

.text

```
main:      li $v0, 8
           la $a0, string1
           li $a1, 100
           syscall
           li $v0, 4
           la $a0, string1
           syscall
```

d)

```
                .data
                .text
main:           li $v0, 5
                syscall
                move $a1, $v0
                li $v0, 5
                syscall
                move $a2, $v0
                li $v0, 1
                add $a0, $a1, $a2
                syscall
```