

MIPS (BUỔI 1)

GIỚI THIỆU ASSEMBLY

Link tải phần mềm biên dịch (tải từ classroom, hoặc search từ google [tại đây](#))

1/ Program Template

VD1: Ngôn ngữ C

```
#include <stdio.h>
#include "mylib.h"

<Khai báo các thư viện>
main()
{
    printf (...);
    scanf();

    luythua(3,2);
}
```

VD2: Ngôn ngữ Pascal

```
Program <ten chương trình>
Var <khai báo các biến>
Begin
    <Code here>;
End;
```

Assembly Template:

```
.data
    <khai báo dữ liệu ở đây>

.text
.globl main
.ent main
main:
    <Code here>

.end main
```

2/ Cách khai báo DATA

- Cú pháp:

<Tên biến>: .<KiểuData> <Giá trị>

Một số kiểu Data thường gặp:

.byte	8 bits ($2^8 = 256$ giá trị)
.half	16 bits (2^{16} giá trị)
.word	32 bits
.ascii	ASCII String
.float	32 bits (số thực)
.double	64 bits (số thực)
.space <n>	khởi tạo bộ nhớ <n> byte

Dấu chú thích: '#'

VD: Khai báo biến các kiểu dữ liệu sau

num:	.word	0	# tương đương num = 0
wnum:	.word	50	# tương đương wnum = 50
bnum:	.byte	5	
wketqua:	.word	0	

Cách Gán giá trị:

VD:

Num = 27 (Dành cho ngôn ngữ bậc cao)

Assembly viết như sau:

li \$t0, 27 # \$t0 = 27 (load immediate)

sw \$t0, num # num = 27 (store word)

Tôi muốn gán: wketqua = num

lw \$t0, num # \$t0 = num (load word)

sw \$t0, wketqua # wketqua = num

3. Chỉ dẫn LOAD / STORE

- Cú pháp 1:

Load giá trị **số nguyên** từ memory vào thanh ghi đích

l<type> Thanhghi_đích, memory

Load giá trị **số thực** từ memory vào thanh ghi đích

l.<type> Thanhghi_đích, memory

- Cú pháp 2:

Load giá trị **số nguyên** cụ thể vào thanh ghi đích

li Thanhghi_đích, immediate

Load giá trị **số thực** cụ thể vào thanh ghi đích

li.<type> Thanhghi_đích, immediate

- Cú pháp 3:
Load địa chỉ (address) của memory vào thanh ghi đích
la Thanhghi_đích, memory
- Cú pháp 4:
Lưu trữ (store) nội dung **chứa giá trị số nguyên** của thanh nguồn vào trong memory
s<type> Thanhghi_nguồn, memory

Lưu trữ (store) nội dung **chứa giá trị số thực** của thanh nguồn vào trong memory
s.<type> Thanhghi_nguồn, memory

4. Các biểu thức Số học

- Cú pháp:
Phép cộng
Thanhghi_đích = Thanhghi_nguồn + Src or Immediate
add Thanhghi_đích, Thanhghi_nguồn, Src
- # Phép trừ
Thanhghi_đích = Thanhghi_nguồn - Src or Immediate
sub Thanhghi_đích, Thanhghi_nguồn, Src
- # Phép nhân
Thanhghi_đích = Thanhghi_nguồn * Src or Immediate
mul Thanhghi_đích, Thanhghi_nguồn, Src
- # Phép chia
Thanhghi_đích = Thanhghi_nguồn / Src or Immediate
div Thanhghi_đích, Thanhghi_nguồn, Src
- # Phép chia lấy dư
Thanhghi_đích = Thanhghi_nguồn % Src or Immediate
rem Thanhghi_đích, Thanhghi_nguồn, Src
- # Phép lấy giá trị tuyệt đối
Thanhghi_đích = |Thanhghi_nguồn|
abs Thanhghi_đích, Thanhghi_nguồn

Ví dụ:

wnum1:	.word	651
wnum2:	.word	42
wans1:	.word	0
wans2:	.word	0
wans3:	.word	0
wans4:	.word	0

Để Thực hiện tính toán cơ bản:

wans1 = wnum1 + wnum2

wans2 = wnum1 - wnum2

wans3 = wnum1 * wnum2

wans4 = wnum1 % wnum2

Ngôn ngữ Assembly thực hiện như sau:

wans1 = wnum1 + wnum2

lw \$t0, wnum1

lw \$t1, wnum2

add \$t2, \$t0, \$t1 # \$t2 = \$t0 + \$t1

sw \$t2, wans1 # wans1 = \$t2

Tương tự Phép cộng, trừ nhân chia ⇒ các em tự về nhà làm.

5. Các lời gọi hệ thống (SYSCALL)

VD: Pascal muốn xuất ra màn hình thì dùng lệnh: Write(“chuoi”);

Ngôn ngữ C muốn xuất ra màn hình thì dùng lệnh: printf(“chuoi”);

Ngôn ngữ Assembly thì phải tuân theo các hướng dẫn sau:

Service	System call code	Arguments	Result
print_int	1	\$a0=integer	
print_float	2	\$f12=float	
print_double	3	\$f12=double	
print_string	4	\$a0=string	
read_int	5		integer (in \$v0)
read_float	6		float (in \$f0)
read_double	7		double (in \$f0)
read_string	8	\$a0=buffer, \$a1=length	
sbrk	9	\$a0=amount	
exit	10		

Ghi nhớ:

- Read a integer number

li \$v0, 5 *# load code 5 vào \$v0 để hệ thống nhận biết nhập số nguyên*
 syscall *# Gọi hệ thống để nhập số nguyên và lưu số vừa nhập vào thanh ghi \$v0*

- **Read a float**

li \$v0, 6 *# load code 6 vào \$v0 để hệ thống nhận biết nhập số float*
 syscall *# Gọi hệ thống để nhập số float và lưu số vừa nhập vào thanh ghi \$f0*

- **Read a double**

li \$v0, 7 *# load code 7 vào \$v0 để hệ thống nhận biết nhập số double*
 syscall *# Gọi hệ thống để nhập số double và lưu số vừa nhập vào thanh ghi \$f0*

- **Print a integer**

li \$v0, 1 *# load code 1 vào \$v0 để hệ thống nhận biết xuất số nguyên*
 lw \$a0, number *# Load số cần xuất vào thanh ghi \$a0*
 syscall *# Gọi hệ thống để xuất giá trị chứa trong thanh ghi \$a0 ra màn hình*

- **Print a float**

li \$v0, 2 *# load code 2 vào \$v0 để hệ thống nhận biết xuất số float*
 mov.s \$f12, \$f0 *# Di chuyển giá trị trong \$f0 vào \$f12*
 syscall *# Gọi hệ thống để xuất giá trị float chứa trong thanh ghi \$f12 ra màn hình*

- **Print a double**

li \$v0, 3 *# load code 3 vào \$v0 để hệ thống nhận biết xuất số double*
 mov.s \$f12, \$f0 *# Di chuyển giá trị trong \$f0 vào \$f12*
 syscall *# Gọi hệ thống để xuất giá trị double chứa trong thanh ghi \$f12 ra màn hình*

- **Print a string**

li \$v0, 4 *# load code 4 vào \$v0 để hệ thống nhận biết xuất chuỗi*
 la \$a0, msg *# Load địa chỉ của chuỗi cần xuất vào thanh ghi \$a0*
 syscall *# Gọi hệ thống để xuất giá trị chuỗi chứa trong thanh ghi \$a0 ra màn hình*

- **Exit**

li \$v0, 10 *# load code 10 vào \$v0 để hệ thống nhận biết thoát chương trình*
 syscall *# Gọi hệ thống để thoát chương trình*

Ví dụ 1: Viết chương trình Xuất một chuỗi "Xin chào các bạn" ra màn hình

(Mở phần mềm MARS, Soạn code, và lưu lại với file có đuôi .asm, VD: "vidu1.asm")

```
.data
    input:      .asciiz      "Xin chao cac ban"

.text
.globl main
main:
    # print input
    li $v0, 4          # Load code =4 vào thanh ghi $v0 để hệ thống biết print chuỗi
```

<i>la \$a0, input</i>	<i># Load địa chỉ của chuỗi input cần xuất vào \$a0</i>
<i>syscall</i>	<i># Gọi hệ thống để xuất nội dung trong \$a0 ra</i>
<i># exit</i>	
<i>li \$v0, 10</i>	<i># Load code =10 vào thanh ghi \$v0 để hệ thống biết là exit</i>
<i>syscall</i>	<i># Gọi hệ thống để exit</i>

Ví dụ 2: Viết chương trình nhập vào 2 số nguyên. Tính tổng và xuất kết quả

```
.data
    msg: .asciiz "Nhap 2 so nguyen de tinh tong, cach nhau boi Enter: \n"
.text
.globl main
main:
    # Print string msg
    li $v0, 4
    la $a0, msg
    syscall

    # Nhap so nguyen thu nhat
    li $v0, 5      # Load code =5 vào thanh ghi $v0 để hệ thống biết là nhập số nguyên
    syscall        # Gọi hệ thống để cho phép nhập số nguyên, và lưu số vừa nhập vào $v0
    move $t1, $v0  # Di chuyển giá trị của thanh ghi $v0 vào $t1 ($t1 = số thứ 1)

    #Nhap so nguyen thu hai
    li $v0, 5      # Load code =5 vào thanh ghi $v0 để hệ thống biết là nhập số nguyên
    syscall        # Gọi hệ thống để cho phép nhập số nguyên, và lưu số vừa nhập vào $v0
    move $t2, $v0  # Di chuyển giá trị của thanh ghi $v0 vào $t2 ($t2=số thứ 2)

    # Tính tổng
    add $t3, $t1, $t2    # $t3 = $t1+$t2

    #print a interger
    li $v0, 1          # Load code=1 để hệ thống nhận biết print a integer
    move $a0, $t3       # Di chuyển giá trị trong $t3 vào $a0 để xuất ra
    syscall             # Gọi hệ thống để xuất giá trị trong $a0 ra ngoài

    # Thoát
    li $v0, 10
    syscall
```

Bài tập thực hành:

- 1/ Viết chương trình in ra màn hình câu "Sinh viên TDTU xin chào các bạn"
- 2/ Viết chương trình nhập vào hai số nguyên từ bàn phím, tính **hiệu**, **tích** của 2 số nguyên đó.
- 3/ Viết chương trình nhập vào một số nguyên từ bàn phím, tính bình phương của số nguyên đó.
- 4/ Viết chương trình nhập vào 2 số thực từ bàn phím, tính tổng 2 số thực đó

5/ Viết chương trình nhập vào 2 số nguyên x, y từ bàn phím, tính thương 2 số nguyên đó. (Lưu ý: Thương của 2 số x/y là một số thực)

6/ Viết chương trình tính diện tích, chu vi hình chữ nhật và xuất kết quả ra màn hình.