

Arquitetura e Organização de Computadores

Universidade do Estado de Mato Grosso Faculdade de
Ciências Exatas e Tecnológicas

Prof Me. João Ricardo dos Santos Rosa

joao.santos@unemat.br

2023/02

ASSEMBLY MIPS – PONTO FLUTUANTE

Números Reais

- Números representados com casas decimais
 - **Float**
 - 32bits
 - Um único registrador do coprocessador 1
 - **Double**
 - 64 bits
 - Dois registradores do coprocessador 1

Números Reais - float

Registers	Coproc 1	Coproc 0
Name	Float	Double
\$f0	0.0	0.0
\$f1	0.0	
\$f2	0.0	0.0
\$f3	0.0	
\$f4	0.0	0.0
\$f5	0.0	
\$f6	0.0	0.0
\$f7	0.0	
\$f8	0.0	0.0
\$f9	0.0	
\$f10	0.0	0.0
\$f11	0.0	
\$f12	0.0	0.0
\$f13	0.0	
\$f14	0.0	0.0
\$f15	0.0	
\$f16	0.0	0.0
\$f17	0.0	
\$f18	0.0	0.0
\$f19	0.0	
\$f20	0.0	0.0
\$f21	0.0	
\$f22	0.0	0.0
\$f23	0.0	
\$f24	0.0	0.0
\$f25	0.0	
\$f26	0.0	0.0
\$f27	0.0	
\$f28	0.0	0.0
\$f29	0.0	
\$f30	0.0	0.0
\$f31	0.0	

Impressão de Números Reais

- Para lermos um numero real precisamos declarar uma variável atribuindo um numero do tipo float.
- Sendo assim, em **.data** declaramos a variavel e seu tipo :
 - Usamos **.float** para definirmos uma variável do tipo **float**
 - Colocamos o numero 33.254 para ser atribuído a variavel
- Em **.text**:
 - Realizamos a intrução **\$v0,2** para imprimir um float
 - Vimos nas aulas passadas que:
 - **\$v0,1** imprime um inteiro
 - **\$v0,4** imprime uma string
 - **\$v0,5** lê um numero inteiro
 - **\$v0,8** lê uma string
- Ao dar o comando syscal o valor deve ficar armazenado em **\$f12**.

Exemplo

.data

```
num1 .float 33.254 # numero do tipo float atribuído a variável num1
```

.text

```
li $v0,2 #Sistema, prepara par imprimir um float
```

```
lwc1 $f12, num1 #no caso dos float, os registradores estão no co-  
processador 1 (cp1)
```

```
#Sempre devemos colocar o float em $f12, ou o valor correto  
não é impresso
```

```
Syscall
```

```
li $v0, 10 #comando para finalizar o programa
```

```
syscall
```

Exemplo

- Escreva um programa que imprima o peso de duas pessoas para o usuário

Caro usuário o peso da 1ª pessoa é _____

E o peso da 2ª pessoa é _____

Exemplo

.data

```
msg1: .ascii "Caro usuário o peso da 1° pessoa é: \n"
num1: .float 75.6
msg2: .ascii "Caro usuário o peso da 2° pessoa é: \n"
num2: .float 75.6
```

.text

```
li $v0,4 #Sistema, prepara para imprimir uma string
la $a0, msg1
syscall
```

```
li $v0,2 #Sistema, prepara par imprimir um float
lwc1 $f12, num1 #no caso dos float, os registradores estão no co-processador 1
(cp1)
syscall
```

```
li $v0,4 #Sistema, prepara par imprimie uma string
la $a0, msg2
syscall
```

```
li $v0,2 #Sistema, prepara par imprimie um float
lwc1 $f12, num2 #no caso dos float, os registradores estão no co-processador 1
(cp1)
syscall
```

```
li $v0,10 #Encerrando o programa
syscall
```

Exemplo

- Escreva um programa que leia o nome do corredor, depois imprima o nome do corredor e seu tempo de corrida.

Qual o nome do atleta? _____

O nome do atleta __XXXX__

Seu tempo de corrida é de 1.53

.data

msg: .ascii "Qual o nome do atleta? "
nome: .space 50
msg2: .ascii "O nome do atleta é "
msg3: .ascii "O tempo de corrida do atleta

é de "

tempo: .float 1.53

.text

#imprimir msg

li \$v0,4
la \$a0, msg
syscall

#leitura do nome

li \$v0,8 #comando para ler uma string
la \$a0, nome
la \$a1, 50
syscall

#imprimir msg2

li \$v0,4
la \$a0, msg2
syscall

#impressao do nome

li \$v0,4
la \$a0, nome
syscall

imprimir msg 3

li \$v0,4
la \$a0, msg3
syscall

#Sistema, prepara par imprimir um float

li \$v0,2
lwc1 \$f12, tempo #no caso dos float, os registradores
estão no co-processador 1 (cp1)
syscall

Atividade

- Escreva um programa em MIPS que mostre o nome e o preço de 4 carnes no açougue

AÇOUGUE HOMER SIMPSOM

O valor da carne moída é 22.90

O valor do coxão mole é 33.80

O valor da picanha é 62.40

O valor da fraldinha é 45.50

Leitura de Números Reais

- Para lermos um numero real precisamos
 - Realizamos a intrução **\$v0**,6 para imprimir um float
 - Quando fizer uma chamada de sistema (**syscall**) O valor lido como float vai para **\$f0**.
 - Mas como vimos anteriormente para imprimirmos o numero real o valor precisa estar em **\$f12**
 - Sendo assim precisamos mover o conteúdo de **\$f0** para **\$f12**. Para isso utilizamos a instrução **mov.s**

Exemplo

.data

msg1: .asciiz "Digite um numero real "

msg2: .asciiz "O numero real digitado foi "

.text

#impressao da msg1

li \$v0,4

la \$a0, msg1

syscall

#leitura numero real;

li \$v0,6

syscall #ao dar syscall o conteudo vai para \$f0

#impressao da msg2

li \$v0,4

la \$a0, msg2

syscall

mov.s \$f12,\$f0 #movendo o conteudo de \$f0 para dentro de \$f12

li \$v0,2 #comando para imprimir um numero real

syscall

#impressao da msg

li \$v0,10

syscall

Exemplo

- Escreva um programa que leia dois números decimais e depois realize a soma entre ambos

Digite o primeiro numero decimal _____

Digite o segundo numero decimal _____

A soma dos números é _____

.data

```
msg1: .asciiz "Forneça o 1° numero decimal\n"  
msg2: .asciiz " Forneça o segundo numero decimal\n"  
msg3: .asciiz " A soma dos numeros é\n "
```

.text

```
#imprimindo mensagem para o usuario
```

```
li $v0, 4
```

```
la $a0, msg1
```

```
syscall
```

```
#lendo o numero do tipo real
```

```
li $v0, 6
```

```
syscall # o valor lido fica salvo em $f0
```

em \$f12

```
# para imprimir um numero real esse numero precisa estar
```

```
# o comando mov $f12,$f0 da erro em COP1
```

```
# temos que usar o comando mov.s
```

```
mov.s $f3, $f0
```

```
li $v0, 4
```

```
la $a0, msg2
```

```
syscall
```

```
#lendo o numero do tipo real
```

```
li $v0, 6
```

```
syscall
```

```
mov.s $f4, $f0
```

```
add.s $f12, $f3, $f4
```

```
li $v0, 4
```

```
la $a0, msg3
```

```
syscall
```

```
li $v0, 2 #comando para imprimir um numero real  
syscall #lendo o numero do tipo real ao imprimir  
sera impresso oq estiver dentro do f12
```

```
li $v0, 10
```

```
syscall #comando para finalizar o programa
```

Atividade

- Escreva um programa o preço de uma compra, o valor dado pelo usuário e imprima o seu troco

Digite o valor da compra _____

Digite o valor entregue

Seu troco é: _____

Números Reais Double

Double

- 64 bits
- Dois registradores do coprocessador 1
- Em double só trabalhamos com registradores pares. Sendo assim, precisamos atribuir sempre uma constante

Números Reais – double

Registers	Coproc 1	Coproc 0
Name	Float	Double
\$f0	0.0	0.0
\$f1	0.0	
\$f2	0.0	0.0
\$f3	0.0	
\$f4	0.0	0.0
\$f5	0.0	
\$f6	0.0	0.0
\$f7	0.0	
\$f8	0.0	0.0
\$f9	0.0	
\$f10	0.0	0.0
\$f11	0.0	
\$f12	0.0	0.0
\$f13	0.0	
\$f14	0.0	0.0
\$f15	0.0	
\$f16	0.0	0.0
\$f17	0.0	
\$f18	0.0	0.0
\$f19	0.0	
\$f20	0.0	0.0
\$f21	0.0	
\$f22	0.0	0.0
\$f23	0.0	
\$f24	0.0	0.0
\$f25	0.0	
\$f26	0.0	0.0
\$f27	0.0	
\$f28	0.0	0.0
\$f29	0.0	
\$f30	0.0	0.0
\$f31	0.0	

Impressão de Números double

- Para lermos e imprimirmos um numero real do tipo double precisamos declarar uma constante do tipo double = 0
- Sendo assim, em **.data** declaramos a variavel e seu tipo :
 - Usamos **.double** para declarar uma variável constante
 - Para esta variável, atribuímos o valor 0,0
- Em **.text**:
 - Realizamos a intrução **\$v0,3** para imprimir um double e **\$v0,7** para ler um double
- Ao dar o comando syscal o valor deve ficar armazenado em **\$f12**.

Exemplo

- Escreva um programa em que o usuário forneça um numero decimal e em seguida o imprima na tela o numero fornecido

Forneça um numero decimal _____

O numero fornecido foi _____

•data

```
msg: .asciiz "forneça um numero decimal:  "  
msg2: .asciiz "o numero fornecido foi:  "  
zero: .double 0.0
```

•text

```
# Imprimindo a mensagem para o usuario
```

```
li $v0, 4  
la $a0, msg1  
syscall
```

```
# Lendo o numero
```

```
li $v0, 7  
syscall # o valor lido estará salvo em $f0
```

```
ldc1 $f2, zero  
add.d $f12, $f2, $f0
```

```
# Imprimindo a mensagem para o usuario
```

```
li $v0, 4  
la $a0, msg1  
syscall
```

```
# imprimindo o numero
```

```
li $v0, 3  
syscall
```