

# 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](mailto:joao.santos@unemat.br)

2024/01

# ASSEMBLY MIPS

# Leitura de Inteiros

- Para lermos um numero Inteiro precisamos mandar uma mensagem para o usuário solicitando para digitar um valor
- Sendo assim, em **.data** declaramos a string e seu tipo :
  - Usamos **.ascii** para definirmos uma variável do tipo string
  - Colocamos entre **"XXX "** o texto a ser apresentado
- Em **.text**:
  - Realizamos a intrução **\$v0,5** para ler um inteiro
  - Vimos nas aulas passadas que:
    - **\$v0,1** imprime um inteiro
    - **\$v0,4** imprime uma string
- Ao dar o syscal o comando fica armazenado em **\$a0**.

# Exemplo

- Escreva um programa que apresente uma mensagem de saudação, leia a idade e apresente a idade digitada para o usuário

Olá digite sua idade \_\_\_\_\_  
Sua idade é \_\_\_\_\_

# Exemplo

**.data**

**saudacao:** **.asciiz** "Ola por favor digite sua idade " # primeira mensagem a ser mostrada  
**saida:** **.asciiz** "sua idade e: " #segunda mensagem

**.text**

**li \$v0,4** #comando para imprimir uma string

**la \$a0, saudacao** # comando para mover da memoria principal para o registrador a0  
**syscall** # imprimo a variavel saudacao que esta no registrador a0

**li \$v0,5** #comando para ler um inteiro

**syscall** #quando der enter o valor sera armazenado em v0

**move \$t0, \$v0** #movi o registrador v0 para t0 pois vou usar o v0 para  
#chamar uma string abaixo

**li \$v0,4** #comando para imprimir uma string que agora esta sem nada

**la \$a0, saida** # comando para mover da memoria principal para o registrador a0  
**syscall** # imprimo a variavel saudacao que esta no registrador a0

**li \$v0,1** #comando para imprimir um inteiro

**move \$a0, \$t0** #movi o registrador v0 para t0 pois vou usar o v0 para  
#chamar um numero inteiro abaixo

**syscall** # imprimo o valor inteiro

**li \$v0, 10,** #comando para finalizar o programa  
**syscall**

# Exemplo

- Escreva um programa que leia dois valores e realize a soma entre eles

**Digite o primeiro numero**\_\_\_\_\_

**Digite o segundo numero** \_\_\_\_\_

\_\_\_\_\_

.data

.text

soma1: .asciiz "Digite o primeiro num " # primeira mensagem a ser mostrada

soma2: .asciiz "Digite o segundo num: " #segunda mensagem

li \$v0,4 #comando para imprimir uma string

la \$a0, soma1 # comando para mover da memoria principal para o registrador a0

syscall # imprimo a variavel soma1 que esta no registrador a0

li \$v0,5 #comando para ler um inteiro

syscall #quando der enter o valor sera armazenado em v0

move \$t0, \$v0 #movi para o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo

li \$v0,4 #comando para imprimir uma string que agora esta sem nada

la \$a0, soma2 # comando para mover da memoria principal para o registrador a0

syscall # imprimo a variavel saudacao que esta no registrador a0

li \$v0,5 #comando para ler um inteiro

syscall #quando der enter o valor sera armazenado em v0

move \$t1, \$v0 #movi o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo

add \$t3,\$t1,\$t0

li \$v0,1 #comando para imprimir um inteiro

move \$a0, \$t3 #movi o registrador v0 para t0 pois vou usar o v0 para

#chamar um numero inteiro abaixo abaixo

syscall # imprimo o valor inteiro

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

syscall

# Atividade

- Escreva um programa em MIPS que leia dois números de telefones celulares e depois apresente os telefones digitados

Digite o N° do primeiro telefone celular \_\_\_\_\_

O numero digitado foi \_\_\_\_\_

Digite o N° do segundo telefone celular \_\_\_\_\_

O numero digitado foi \_\_\_\_\_

**.data**

```
telefone1: .asciiz "Digite o N° do primeiro telefone "  
saida1: .asciiz "O numero digitado foi "  
saida2: .asciiz "O numero digitado foi "  
telefone2: .asciiz "Digite o N° do segundo telefone "
```

**.text**

```
li $v0,4 #comando para imprimir uma string  
la $a0, telefone1 # comando para mover da memoria principal para o registrador a0  
syscall # imprimo a variavel soma1 que esta no registrador a0  
  
li $v0,5 #comando para ler um inteiro  
syscall #quando der enter o valor sera armazenado em v0  
  
move $t0, $v0 #movi para o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo  
li $v0,4 #comando para imprimir uma string que agora esta sem nada  
la $a0, saida1 # comando para mover da memoria principal para o registrador a0  
syscall # imprimo a variavel saudacao que esta no registrador  
  
li $v0,1 #comando para imprimir um inteiro  
move $a0, $t0 #movi o registrador v0 para t0 pois vou usar o v0 para chamar um numero inteiro  
  
syscall # imprimo o valor inteiro  
  
li $v0,4 #comando para imprimir uma string  
la $a0, telefone2 # comando para mover da memoria principal para o registrador a0  
syscall # imprimo a variavel soma1 que esta no registrador a0  
  
li $v0,5 #comando para ler um inteiro  
syscall #quando der enter o valor sera armazenado em v0
```

abaixo



**move \$t1, \$v0** #movi para o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo  
**li \$v0,4** #comando para imprimir uma string que agora esta sem nada  
**la \$a0, saida2** # comando para mover da memoria principal para o registrador a0  
**syscall** # imprimo a variavel saudacao que esta no registrador a0

**li \$v0,1** #comando para imprimir um inteiro  
**move \$a0, \$t1** #movi o registrador v0 para t0 pois vou usar o v0 para  
#chamar um numero inteiro abaixo  
**syscall** # imprimo o valor inteiro

**li \$v0, 10**, #comando para finalizar o programa  
**syscall**

# Atividade

- Escreva um programa em MIPS que leia dois números realize a soma de ambos e depois receba um outro numero que será subtraído do valor da soma

**Digite o primeiro numero** \_\_\_\_\_

**Digite o segundo numero** \_\_\_\_\_

**Digite o terceiro numero** \_\_\_\_\_

**O numero digitado foi** \_\_\_\_\_

## .data

```
num1: .ascii "Digite o 1° numero "  
num2: .ascii "Digite o 2° numero "  
num3: .ascii "Digite o 3° numero "
```

## .text

```
li $v0,4 #comando para imprimir uma string  
la $a0, num1 # comando para mover da memoria principal para o registrador a0  
syscall # imprimo a variavel soma1 que esta no registrador a0  
  
li $v0,5 #comando para ler um inteiro  
syscall #quando der enter o valor sera armazenado em v0  
  
move $t0, $v0 #movi para o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo  
  
li $v0,4 #comando para imprimir uma string que agora esta sem nada  
la $a0, num2 # comando para mover da memoria principal para o registrador a0  
syscall # imprimo a variavel saudacao que esta no registrador a0  
  
li $v0,5 #comando para ler um inteiro  
syscall #quando der enter o valor sera armazenado em v0  
  
move $t1, $v0 #movi o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo  
  
add $t2,$t1,$t0  
  
li $v0,4 #comando para imprimir uma string  
la $a0, num3 # comando para mover da memoria principal para o registrador a0  
syscall # imprimo a variavel soma1 que esta no registrador a0
```

li \$v0,5 #comando para ler um inteiro

syscall #quando der enter o valor sera armazenado em v0

move \$t3, \$v0 #movi o registrador v0 para t0 pois vou usar o v0 para chamar uma string abaixo

sub \$t4, \$t2,\$t3

li \$v0,1 #comando para imprimir um inteiro

move \$a0, \$t4 #movi o registrador v0 para t0 pois vou usar o v0 para

#chamar um numero inteiro abaixo abaixo

syscall # imprimo o valor inteiro

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

syscall

# *Leitura de Strings*

- Em **.data**, declaramos a string e o seu tamanho
  - Usamos **.space** para determinar o tamanho da variável
- Em **.text**
  - Realizamos a instrução **li \$v0,8**
  - **la \$a0, <variavelRam>**
  - **la \$a1, <tamanhoDaVariavel>**
- Ao dar o **syscall** o comando fica armazenado em **\$a0**.

# Exemplo

- Escreva um programa que lê um nome e o imprima, no formato abaixo

Qual é o seu nome? \_\_\_\_\_

Olá XXXX

## .data

```
pergunta: .asciiiz "Qual seu nome  "  
saudacao: .asciiiz "Ola  "  
nome: .space 50
```

## .text

```
#impressao da pergunta
```

```
li $v0,4
```

```
la $a0, pergunta
```

```
syscall
```

```
#leitura do nome
```

```
li $v0, 8 #comando para ler uma string
```

```
la $a0, nome
```

```
la $a1, 25
```

```
syscall
```

```
#impressao da saudacao
```

```
li $v0,4
```

```
la $a0, saudacao
```

```
syscall
```

```
#impressao do nome
```

```
li $v0,4
```

```
la $a0, nome
```

```
syscall
```

# Atividade

- Escreva um programa que leia o nome e depois leia o nome da rua

Qual é o seu nome? \_\_\_\_\_

Qual o nome da sua rua ? \_\_\_\_\_

Prazer XXXXX

Você mora na rua : XXXX



**.data**

```
pergunta: .ascii "Qual seu nome "  
pergunta2: .ascii "Qual o seu endereco? "  
saudacao: .ascii "Ola "  
afirmacao: .ascii "voce mora no seguinte endereco "  
nome: .space 50  
endereco: .space 50
```

**.text**

```
#impressao da pergunta nome
```

```
li $v0,4
```

```
la $a0, pergunta
```

```
syscall
```

```
#leitura do nome
```

```
li $v0, 8 #comando para ler uma string
```

```
la $a0, nome
```

```
la $a1, 25
```

```
syscall
```

```
#impressao da pergunta endereco
```

```
li $v0,4
```

```
la $a0, pergunta2
```

```
syscall
```

#leitura do endereco

li \$v0, 8

la \$a0, endereco

la \$a1, 25

syscall

#impressao da saudacao

li \$v0,4

la \$a0, saudacao

syscall

#impressao do nome

li \$v0,4

la \$a0, nome

syscall

#impressao da afirmacao

li \$v0,4

la \$a0, afirmacao

syscall

#impressao do endereco

li \$v0,4

la \$a0, endereco

syscall

# Atividade

- Escreva um programa que leia o Loguin e senha e depois imprima uma mensagem de saudação com base no login apresentando a senha

Loguin \_\_\_\_\_

Senha \_\_\_\_\_

Bem vindo **XXXXXX**

Sua senhas é **XXXX**

.data

```
msg1: .asciiz " Qual seu loguin ?"  
msg2: .asciiz " Qual sua senha ?"  
msg3: .asciiz " Prazer "  
msg4: .asciiz " Sua senha é "  
log: .space 23  
senha: .space 40
```

.text

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

```
li $v0,8  
la $a0, log  
la $a1 23  
syscall
```

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

```
li $v0,8  
la $a0, senha  
la $a1, 40  
syscall
```

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

li \$v0, 4  
la \$a0, msg4  
syscall

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

li \$v0, 10  
syscall