

# Aula 4 - Estrutura do programa MIPS

---

(11/nov)

- Estrutura de um programa em assembly MIPS
  - Pseudoinstruções úteis
  - Syscalls
- 

## Estrutura de um programa:

---

```
.data
    # declarações de variáveis e constantes
.text
    # código MIPS
main:
    # programa principal
```

## Tipos de dados:

---

Tipos	nome	dados
.word	w1, ... , wn	dado de 32 bits (palavra)
.half	h1, ... , hn	dado de 16 bits (meia palavra)
.byte	b1, ... , bn	dado de 8 bits
.ascii	str	cadeia de caracteres
.asciiz	str	cadeia de caracteres terminando com o caracter nulo (\0)

## System calls

---

Executar tarefas que dependam do sistema operacional

- Alocar memória
- Lidar com entrada e saída

Chamada do programa em baixo nível para o sistema operacional para lidar com entrada e saída

Para fazer uma syscall, é necessário:

1. Carregar o código da syscall no registrador \$v0
2. Carregar os argumentos necessários em \$a0-\$a3 (pode ter argumentos ou não)

3. Fazer a chamada usando a instrução "syscall"

Serviço	Código	Argumentos	Resultado
Imprimir inteiro	1	\$a0 = inteiro	n.a.
Imprimir uma string	4	\$a0 = endereço da string	n.a.
ler um inteiro	5	n.a	\$v0 = valor lido
ler uma string	8	\$a0 = endereço da string e \$a1 = qtde. de caracteres +1	n.a.
Alocar memória	9	\$a0 = número de bytes	\$vo = endereço do bloco
<b>Encerrar um programa</b>	10	n. a.	n.a.
Imprimir um caracter	11	\$a0 = inteiro(ASCII)	n.a.
Ler um caracter	12	n. a.	\$vo = caracter lido

## Pseudoinstruções

li

- li reg, const
- Passa um registrador e uma constante
- E ela carrega o valor da constante no registrador reg

la

- la reg, rotulo (label)
- Carrega o endereço de memória do rótulo no registrador reg

move

- move reg1, reg2
- Carrega em reg1 o reg2
- reg1 = reg2

## Exemplo: Somar dois números

- Soma
- Syscall leitura e valor calculado

```
.data
```

```
# todas as mensagens que quero exibir na tela
msg1:  .ascii "Digite a: " #msg1 é o rótulo da variavel
msg2:  .ascii "Digite b: "
res:   .ascii "a + b = "
nl:    .ascii "\n"

.text

main:
    # $t0: a
    # $t1: b

    # imprimir a frase "digite a"
    li    $v0, 4      # $v0=4
    la    $a0, msg1   # carrega em $a0 o end. de memória de msg1
    syscall

    # le um numero inteiro do teclado
    li    $v0, 5      # $v0=5
    syscall
    move   $t0, $v0    # move para $t0 o valor lido do teclado que esta
em $v0

    # imprimir a frase "digite b"
    li    $v0, 4      # $v0=4
    la    $a0, msg2   # carrega em $a0 o end. de memória de msg1
    syscall

    # le um numero inteiro do teclado
    li    $v0, 5      # $v0=5
    syscall
    move   $t1, $v0    # move para $t0 o valor lido do teclado que esta
em $v0

    add    $t2, $t0, $t1 # $t2 = $t0 + $t1
    # Imprime o resultado na tela
    li    $v0, 4      # $v0=4
    la    $a0, res    # carrega em $a0 o end. de memória de res
    syscall

    li    $v0, 1
    move   $a0, $t2    # $sa0 = $t2, o numero a ser impresso
    syscall

    # Imprimindo o \n
    li    $v0, 4
    la    $a0, nl
    syscall

    # Encerrar o programa
```

```
li    $v0, 10  
syscall
```

## Para rodar o código

---

1. Abre o terminal na pasta do arquivo (.spim)
2. digite o comando:

```
spim -f arquivo.spim
```