

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACOM - FACULDADE DE COMPUTAÇÃO
SISTEMA DE INFORMAÇÃO

LUCAS ALBINO MARTINS

Matrícula: 12011ECP022

**TRABALHO 02: Desenvolvimento de seu primeiro
programa em Assembly MIPS.**

Disciplina: Arquitetura e Organização de Computadores.

Uberlândia
2020

Escreva o programa no MARS, certifique-se de que ele funciona, e então recorte e cole na sua aba pessoal de anotações de classe. Crie uma "Página" na pasta "Tarefas" chamada "Primeiro Programa".

Desenvolva em assembly um programa que resolva a seguinte equação matemática:

$$x = (y+14) - 64$$

assuma que o valor de y encontra-se no registrador \$s7 e o valor de x deve ser armazenado no registrador \$s6

#-----

#programa1.asm

#

#DDA 22.08.2020

#

#Aluno: Lucas Albino Martins, Matricula 12011ECP022

#Obs: envei naquela area de notas, mas ta aparecendo no teams q não enviei a atividade 2 por isso reenviei na opcao de atraso

#

#DESC: programa simples que resolve uma equação matemática

#

#-----

.data

Criando uma variavel para impressao do valor final.

str1: .asciiz "\n Entre com o valor para y: "

str2: .asciiz "\n O valor de y em \$s7 eh: "

str3: .asciiz "\n O valor da expressao em salvo em \$s6 eh: "

.text

$x = (y+14) - 64$

assumo que o valor de y encontra-se no registrador \$s7

e o valor de x deve ser armazenado no registrador \$s6

imprimindo string para o usuario entrar com um valor de y

li \$v0, 4 # atribui 4 para \$v0. Código para print_str

la \$a0, str1 # carrega endereço de str1 em \$a0

syscall # chamada de sistema para I/O

varrendo o valor digitado no teclado

li \$v0, 5 # atribui 5 para \$v0. Código para read_int

syscall # chamada de sistema para I/O

movendo o valor lido pelo teclado para a variavel

move \$s7, \$v0 # copia conteúdo digitado para \$t2 para preservar
dado

li \$t0, 1 # \$t0 = 1 eh usado como aux.

colocando um valor no registrador \$s7

#add \$s7, \$zero, 60 # s7 = 60+0

imprimindo o valor de y em \$s7

imprimindo a string

li \$v0, 4 # atribui 4 para \$v0. Código para print_str

la \$a0, str2 # carrega endereço de str_neg em \$a0

syscall # chamada de sistema para I/O

imprimindo o resultado

li \$v0, 1 # atribui 1 para \$v0. Código para print_int

move \$a0, \$s7 # copia valor de \$s7 a ser impresso para \$a0

syscall # chamada de sistema para I/O

resolve (y + 14) e atribui o valor a um registrador temporário

addi \$t1, \$zero, 14 # t0 = 14+0

add \$t2, \$s7, \$t1 # t1 = y + 14

```
# resolve (y+14) - 64 e atribui a um registrador temporário
addi      $t3, $zero, 64 # t0 = 64+0

sub       $s6, $t2, $t3   # s6 = (y+14) -64

# imprimindo a string
li        $v0, 4          # atribui 4 para $v0. Código para print_str
la        $a0, str3       # carrega endereço de str_neg em $a0
syscall

# imprimindo o resultado
li        $v0, 1          # atribui 1 para $v0. Código para print_int
move      $a0, $s6        # copia valor de $s6 a ser impresso para $a0
syscall

# mostrando ao sistema operacional que a execução foi finalizada.
li        $v0, 10         # atribui 10 para $v0. Código para exit (termina
programa)
syscall                  # chamada de sistema para I/O
```