

Arquitetura e Organização de Computadores I



MARS

MARS (MIPS Assembler and Runtime Simulator)

<http://courses.missouristate.edu/kenvollmar/mars/>

Hello, World!

PROGRAMA: Hello, World!

```
        .data          # Data declaration section
out_string: .ascii      "\nHello, World!\n"
        .text          # Instruções na linguagem Assembly
main:    # Start of code section
        li $v0, 4       # system call code for printing string = 4
        la $a0, out_string # load address of string to be printed into $a0
        syscall         # call operating system to perform operation in $v0
                        # syscall takes its arguments from $a0, $a1, ...
```

Part I

1. Ler dois valores do Usuário;
2. Somar os dois valores;
3. Exibir os dois valores;

Part II

1. Mostrar a multiplicação (x3) do primeiro valor lido;

Part III

1. Emitir um sinal sonoro de 5 Segundos, junto com a mensagem: “calculando”, antes de exibir o resultado;

Exercício

- Incremente o programa desenvolvido em sala de aula, fazendo com que ele escreva o resultado da soma em um arquivo.

Exercícios (Patterson 5 ed.)

2.1 [5] <§2.2> For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables `f`, `g`, `h`, and `i` are given and could be considered 32-bit integers as declared in a C program. Use a minimal number of MIPS assembly instructions.

`f = g + (h - 5);`

2.2 [5] <§2.2> For the following MIPS assembly instructions above, what is a corresponding C statement?

```
add    f, g, h
```

```
add    f, i, f
```

Perguntas?



See ya!

THANK YOU