

# Trabalho 3 de Organização e Arquitetura de Computadores - Turma C

Marcos Paulo Cayres Rosa (14/0027131)

Departamento de Ciência da Computação  
Organização e Arquitetura de Computadores  
Brasília, Brasil

**Abstract**—Esse documento tem como finalidade demonstrar como foi construído um manipulador de strings no Mars Mips em linguagem Assembly (*Abstract*)

**Keywords**—*mips, mars, simulador, assembly*

## I. OBJETIVOS

Desenvolver um programa de manipulação de strings em *assembly*. Deve interagir com o usuário de forma a executar o que for escolhido e operar em cima do que for fornecido por ele. Para tal é necessário implementar funções que concatenem, modifiquem e determinem o tamanho de strings.

## II. INSTRUÇÕES

Para compilação e execução do programa é preciso seguir os passos descritos abaixo:

1. Utilizando o Mars MIPS, abrir o código fonte 140027131.asm, montar (F3) e rodar (F5).
2. Conferir se o terminal Run I/O está limpo antes do uso e se algum outro programa não estava sendo rodado, inclusive o próprio, de forma incompleta.
3. Ao executar, primeiramente entre com uma string e, então, escolha uma das opções do menu apresentado. Ademais, siga as instruções apresentadas no terminal.

## III. DOCUMENTAÇÃO

Abaixo estão listadas as funções implementadas e como estas foram planejadas:

1. Entrar string

Utiliza-se do “syscall” para ler strings (oitavo modo de operação), tendo como tamanho máximo 100 caracteres.

Ao final, determina o fim da palavra e retira o caracter correspondente (Enter), pois esse é usado pelo usuário para determinar a inserção da string, porém não faz parte desta. Como o tempo de execução dessa tarefa é curto para o tamanho de string especificado, optei por manter essa implementação.

### 2. Caixa alta

Analisa cada caracter e, caso esteja entre “a” e “z”, modifica este para o correspondente em caixa alta, subtraindo o valor de “a” e somando o de “A”. Termina o procedimento ao final da palavra e imprime o resultado obtido.

### 3. Caixa baixa

Mesmo processo que o de caixa alta, porém analisando entre “A” e “Z”, subtraindo “A” e somando “a”.

### 4. Tamanho

Usa um registrador como contador e a cada ciclo aumenta este. Um ciclo corresponde a leitura de um caracter (byte) distinto, passando para o próximo até que encontre o fim da palavra e, após isso, imprimindo o tamanho.

### 5. Conta caracteres

Mesma implementação para determinar o tamanho com o acréscimo de algumas condições: para o contador aumentar, o caracter atual deve estar entre “0” e “9”, “a” e “z” ou “A” e “Z”.

### 6. Conta dígitos

Possui a mesma codificação da contagem de caracteres, porém só com a condição de estar entre “0” e “9”.

#### 7. Concatena

Determina o final da palavra e coloca para a leitura de string por “syscall” ser executada e salva nessa posição. Ademais, tem as mesmas características da leitura de string.

#### 8. Encerra

Chama o “syscall” (décimo modo de operação) para encerrar a execução do programa.