



Universidade Federal de Pelotas
Centro de Desenvolvimento Tecnológico
Bacharelado em Ciência da Computação
Engenharia de Computação

Arquitetura e Organização de Computadores I

EXERCÍCIOS

Programando no MIPS: Avançado
Assembly Estendido, Strings e Vetores,
Chamadas de Sistema

Prof. Luciano Agostini
agostini@inf.ufpel.edu.br

AOC-1: Trabalho Prático 7

► Instruções

- Use apenas instruções vistas em aula até agora (slides)
- No MARS, use a seguinte configuração:
 - No menu *Settings*, habilite a opção:
 - *Permit extended (pseudo) instructions and formats* e
 - No menu *Settings*, desabilite a opção:
 - *Delayed Branching*

AOC-1: Trabalho Prático 7

► Exercício:

1. Escreva um programa que remova os espaços de uma string. Por exemplo, a entrada

`.data`

```
string: .ascii "Gosto muito do meu professor de AOC-I."
```

deve produzir a string

```
"GostomuitodomeuprofessordeAOC-I."
```

Use apenas uma string (não use uma string de saída ou uma string auxiliar no seu programa). Não esqueça de terminar sua string com nulo (ver tabela ASCII para código do espaço e do `\0` (null)).

A resposta deve ser a string de entrada modificada, e não uma nova string na memória, ou seja, iniciando no endereço de memória `0x10010000`.

AOC-1: Trabalho Prático 7

► Exercício:

2. Escreva um programa que altere uma string para “capitalizar” a primeira letra de cada palavra. Por exemplo, a entrada

```
.data  
    string: .asciiz "meu professor é muito bom"
```

deve produzir a string

"Meu Professor É Muito Bom"

Assuma que a entrada possui apenas espaços e letras minúsculas. Pode haver mais de um espaço entre as palavras.

A resposta deve ser a string de entrada modificada, e não uma nova string na memória, ou seja, iniciando no endereço de memória 0x10010000.

AOC-1: Trabalho Prático 7

► Exercício:

3. Escreva um programa que leia um vetor de 10 posições (.word) da memória (começando na posição 0x10010000) e verifique se o vetor está ou não ordenado. Use o registrador \$t0 como *flag*.
Faça \$t0 = 1 se o vetor estiver ordenado e \$t0 = 0 caso contrário.

AOC-1: Trabalho Prático 7

► Exercício:

4. Escreva um programa que inverta a ordem dos elementos de um vetor (.word) com 5 posições. Por exemplo, a entrada: 1, 2, 3, 4, 5 deve produzir 5, 4, 3, 2, 1.

A resposta deve ser o vetor de entrada modificado, e não um novo vetor na memória, ou seja, iniciando no endereço de memória 0x10010000.

AOC-1: Trabalho Prático 7

► Exercício:

5. Declare três vetores do mesmo tamanho:

```
.data
    tamanho: .word 7
    vetor1:  .word -30, -23, 56, -43, 72, -18, 71
    vetor2:  .word 45, 23, 21, -23, -82, 0, 69
    soma:    .word 0, 0, 0, 0, 0, 0, 0
```

Inicialize um ponteiro para cada vetor (pseudo-instrução `la`) e faça a soma dos elementos dos vetores 2 a 2.

O vetor resultante deve ser armazenado depois dos elementos do segundo vetor.

Exemplo: `soma[i] = vetor1[i]+vetor2[i]`

AOC-1: Trabalho Prático 8

► Exercício:

6. Escreva um programa que repetidamente pergunte ao usuário pelo número de quilômetros viajados e litros de gasolina consumidos e depois imprima o número de quilômetros por litro. Para sair do programa, o usuário deve digitar 0 como número de quilômetros.

Armazene todos os números de quilômetros por litro na memória, iniciando pelo endereço 0x10010000.

Exemplo:

Entrada	Saída	
40		
3	13	-- 13 também deve ser armazenado em 0x10010000
60		
5	12	-- 12 também deve ser armazenado em 0x10010004
0		-- Sai do programa