

USP – ICMC – SSC

SSC0902 – Organização e Arquitetura de Computadores
Programação RISC - MIPS (SPIM / MARS)

Professor responsável: *Fernando Santos Osório***Semestre:** 2019/2**Monitores:** João Vitor e Tiago Marino**Horário:** Seg. 10h10 e Quinta 08h10**E-mail:** fosorio @icmc.usp.br**Wiki:** SSC-902 e **Facebook:** SSC0902-Fosorio**Web:** <http://www.icmc.usp.br/~fosorio/>

LISTA DE EXERCÍCIOS - 03
ASSEMBLY MIPS – Simuladores QtSPIM e MARS

Escreva os programas de acordo com as especificações abaixo, implemente no Emu8086 e teste os programas.

1. Faça um programa que some 2 números pré-definidos e armazenados em **variáveis** (endereços) específicos da memória, denominadas de “**dado1**” e “**dado2**”. Os valores definidos para as variáveis devem ser em **Hexadecimal**, dado1 deve valer 0x01234567 e dado2 deve valer 0x76543210. As duas variáveis guardam **valores INTEIROS de 32 Bits**. Somar os valores, exibir na tela de console o **valor total da soma e salvar em uma variável em memória denominada de “soma”**. Para exibir na tela use a função **SYSCALL** de exibição de valores inteiros na console, termine o programa com a **SYSCALL “Exit”**.
2. Faça um programa similar ao do exercício 1, porém definindo os valores de dado1 e dado2 como 2 valores inteiros em Decimal, dado1 deve valer 01234567 (inteiro em notação decimal de 32 bits) e dado2 deve valer 76543210 (inteiro em notação decimal de 32 bits). Somar, salvar o resultado em “soma”, exibir o resultado e terminar com “exit”.
3. Faça um programa similar ao do exercício 1, porém definindo os valores de dado1 e dado2 como 2 **valores de Ponto Flutuante** (float de precisão simples de 32 bits). Os valores definidos para as variáveis devem ser em **Ponto Flutuante**, dado1 deve valer 12345.12345 e dado2 deve valer 11111.22222. Somar em ponto flutuante, salvar o resultado em “soma” (float 32 bits), exibir o resultado na forma de float (atenção use a syscall que exibe um float!) e terminar com “exit”. Observe a precisão/imprecisão da soma de ponto flutuante como esta que foi realizada, e que possui diversas casas antes e depois da vírgula (float de precisão simples).
4. Faça um programa com um contador que conte de 1 até o número definido por uma constante. Inicialmente você pode contar até 100. Laço de contagem de 1 até o NRO definido (100). Exibir a contagem na tela da console com o respectivo syscall.
5. Faça um programa com um contador que conte de “inicio” até o valor “final” definidos por variáveis em memória. Faça o contador avançar com um passo, definido na variável “incr”. Exibir a contagem na tela da console com o respectivo syscall.

6. Faça um programa com uma contagem regressiva até Zero, que conte de um valor definido em uma variável “início” e com um laço de contagem de “início” até 0. Exibir a contagem na tela da console com o respectivo syscall.
7. Faça um programa em assembly MIPS que imprima na tela triângulos desenhados com o caractere de estrela/asterisco (*) de acordo com um valor pré-definido no código. Exemplo: se o valor for 4, seu programa deverá imprimir na tela da console: “um triângulo, começando com 1, depois 2, depois 3 e por fim 4 asteriscos formando o desenho triangular na tela”. Use a devida função SYSCALL para a exibição dos asteriscos na tela.
- ```
*
**


```
8. Faça o mesmo do exercício acima (exercício 7), porém **o valor agora deverá ser lido do teclado** e o triângulo deve ser apresentado na console de forma invertida (diminuindo o nro. de asteriscos). Lembre-se de usar a devida SYSCALL para ler um número inteiro que representa o número máximo de asteriscos do desenho do triângulo. Exemplo: se o usuário digitar 5, seu programa deverá imprimir:
- ```
*****
****
***
**
*
```
9. Faça um programa que leia 5 entradas (números inteiros) digitados pelo usuário e imprima na tela a soma dos 5 números digitados. Exibir o resultado da soma na tela da console.
10. Faça um programa que procura por um caractere ‘c’ em uma string estática e imprime na tela se encontrou ou não este caractere na string (Mensagens: “achou” / “nao achou”), e indicando em qual posição encontrou o referido caractere (contador inteiro).
11. Faça um programa que recebe uma string como entrada do usuário (digitada pelo usuário) e a imprime na tela.
12. Faça um programa que concatena (une) duas strings (sequências de caracteres), pré-definidas no código e previamente armazenadas em memória, resultando em uma só nova string (como o comando strcat da linguagem “C”). Exibir na tela da console a string nova resultante da concatenação.
13. Faça um programa que leia dois números inteiros (digitados pelo usuário) e calcule a multiplicação destes 2 números. Exibir na console o resultado da multiplicação.
14. Faça um programa que calcule o fatorial de um número informado pelo usuário (digitado pelo usuário). O programa deve calcular o fatorial por multiplicações sucessivas, sem usar recursividade ou chamadas de funções (sub-rotinas). Exibir na console o resultado do cálculo do fatorial.
15. Faça um programa que ordena um vetor estático de números inteiros e o imprime na tela (ordenado).

Listas complementares de exercícios disponível junto a Wiki ICMC da Disciplina SSC0902

[http://wiki.icmc.usp.br/index.php/SSC-902-2019\(FOsorio\)](http://wiki.icmc.usp.br/index.php/SSC-902-2019(FOsorio)) - Listas de Exercícios

Material sobre o MIPS disponível junto a Wiki ICMC da Disciplina SSC0902

[http://wiki.icmc.usp.br/index.php/SSC0902_Material_Aulas_2019\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC0902_Material_Aulas_2019(fosorio)) - Material de Aulas
