**images­Universidade Federal Rural do Semi-Árido - UFERSA**

**Centro de Ciências Exatas e Naturais - CCEN**

**Departamento de Computação - DC**

**Curso Ciência da Computação**

**Disciplina: Sistemas Operacionais**

**Prof. Leiva Casemiro Oliveira**

# 0.1– Tarefa de Sistemas Operacionais

**INSTRUÇÕES:** Essa atividade é em grupo e os código e respostas correspondentes as questões devem ser enviados em um único arquivo (.zip) até a data estabelecida no SIGAA pelo líder do grupo como parte da nota da Unidade em avaliação.

1. Abra o código Assembly a seguir no MARS, execute e veja qual a saída

|  |  |
| --- | --- |
|  | .data  # Vetor 'dados' de bytes  dados: .byte 72, 101, 108, 108, 111, 32, 87, 111, 114, 108, 100, 33, 10  .text  .globl main #deve ser global  main:  lb $a0, dados($0)  #imprimir inteiro  li $v0, 1  syscall  # Terminando o programa  li $v0, 10  syscall |

1. Utilize o código da questão anterior modificando a linha 7 para carregar o endereço da variável “dados” para o registrador $a0 e modificar a linha 10 para imprimir uma string e veja o resultado.
2. Escreva um código em linguagem de montagem para realizar a expressão **a = b + c – (d – e)** sem precisar utilizar mais de 5 registradores.
3. Copie o código da questão 3 e cole no código da questão 2 logo após a diretiva “.text” na linha 4 para utilizá-lo como uma função (utilize o label “func” para identificá-la). Mantenha o código “main” e antes de terminar o programa faça uma chamada a função “func” e imprimindo o resultado. Execute esse código passo-a-passo a ordem que são executadas e os valores dos registradores envolvidos. Explique o que aconteceu.
4. Utilize o mesmo código da questão 4, mas antes de executar no MARS, clique em “Settings -> Initialize Program Counter to global 'main' if defined”. Execute no modo passo-a-passo e explique a diferente da execução da questão 4.
5. Construir um código em assembly do MIPS para executar o seguinte código em alto-nível. Dica: veja o “Help” do MARS as várias opções de instruções e pseudoinstruções para salto.

a = 1;

b = 2;

if (a <= b){

a = a + b;

}else{

a = a – b;

}

1. Modifique o código do item anterior para obter os valores das variáveis “a” e “b” por meio da syscall “read\_int”. No final imprima as variáveis “a” e “b” por meio da syscall “print\_int”.
2. Mostre como o seguinte programa MIPS poderia ser carregado na memória e executado.

# código assembly MIPS

main:

addi $sp, $sp, -4

sw $ra, 0($sp)

lw $a0, x

lw $a1, y

jal diff

lw $ra, 0($sp)

addi $sp, $sp, 4

jr $ra

diff: sub $v0, $a0, $a1

jr $ra

* + 1. Corrija os erros para conseguir compilar
    2. Utilize as opções do MARS para verificar os endereços da tabela de símbolos
    3. Verifique os endereços iniciais e os tamanhos (em bytes e words) dos segmentos de dados e de código (texto).