**UNIVALI – Universidade do Vale do Itajaí**

Curso – Ciência da Computação

Disciplina – Arquitetura e Organização de Processadores

Professor – Douglas Rossi de Melo

Auno – Felipe dos Santos e Nathalia Suzin

**PROGRAMAÇÃO EM LINGUAGEM DE MONTAGEM**

**/ /**

**Exercício 1**

Usando a instrução syscall, implemente um programa que: (1) solicite ao usuário que forneça dois números inteiros (X e Y); (2) realize a soma desses dois valores; e (3) apresente o resultado da soma.

**Em C++:**

#Disciplina: Arquitetura e Organização de Processadores

#Atividade: Avaliação 01 – Programação em Linguagem de Montagem

#Exercício 01

#Aluno: Nathalia Suzin

#include <iostream>

using namespace std;

int main(){

int x, y, soma;

cout << "\nEntre com o valor de x: ";

cin >> x;

cout << "\nEntre com o valor de y: ";

cin >> y;

soma = x + y;

cout << "\nA soma de x e y é igual a: " << soma;

return 0;

}

**Em MIPS:**

#Disciplina: Arquitetura e Organização de Processadores

#Atividade: Avaliação 01 – Programação em Linguagem de Montagem

#Exercício 01

#Aluno: Nathalia Suzin

#Em C++:

#include <iostream>

#using namespace std;

#int main(){

# int x, y, soma;

# cout << "\nEntre com o valor de x: ";

# cin >> x;

# cout << "\nEntre com o valor de y: ";

# cin >> y;

# soma = x + y;

# cout << "\nA soma de x e y é igual a: " << soma;

# return 0;

#}

.data # segmento de dados

str1: .asciiz "\nEntre com o valor de x: " #asciiz = manda mensagem na tela

str2: .asciiz "\nEntre com o valor de y: "

str3: .asciiz "\nA soma de x e y é igual a: "

.text # segmento de código

main:

li $v0, 4 # $v0 indica o código de chamada // 4 = print\_string

la $a0, str1

syscall # mostra a mensagem em str1 // $a0 = argumento

li $v0, 5 # $v0 indica o código de chamada // 5 = read\_int

syscall # litura do valor

add $s1, $v0, $zero #salva o valor de $v0 em $s1 (é somado com zero para não dar divergência no valor)

li $v0, 4

la $a0, str2

syscall # mostra a mensagem em str2

li $v0, 5

syscall

add $s2, $v0, $zero

add $s0, $s1, $s2 # executa a soma dos valores de $s1 e $s2 e salva o resultado em $s0

li $v0, 4

la $a0, str3

syscall # mostra mensagem em str3

li $v0, 1 # $v0 chama o código // 1 = print\_int

add $a0, $s0, $zero # adiciona no argumento $a0 o resultado salvo em $s0 somado com $zero para não dar divergência no valor

syscall # mostra resultado da soma dos valores de $s1 e $s2

#poderia ser feito também:

# li $v0, 4

# la $a0, str3

# syscall

#

# li $v0, 1

# add $a0, $s1, $s2 # a adição é feita anteriormente a chamada do syscall, executando uma instrução a menos

# syscall

▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬

1. Assim que o programa começa a ser executado, a fonte lê o primeiro comando, no endereço 0x00400000, onde foi determinado o tipo da “variável” que será apresentada ou lida (neste caso, o código 4 informa que uma string será mandada para o console de interface). A seguir, é lido o comando de chamada da string, em 0x00400004, e ela é apresentada na tela por meio do syscall.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* É apresentada a mensagem:

Interface gráfica do usuário, Texto, Aplicativo, Word

Descrição gerada automaticamente

1. Então, quando digitado um valor do tipo inteiro, como determinado pelo código 5 (leitura de inteiros), ele será lido e sido salvo no registrador $s1 após o syscall, como indicado no endereço 0x00400018.

Uma imagem contendo Tabela

Descrição gerada automaticamente

1. Os mesmos passos anteriores serão feitos para a leitura do segundo valor, onde será apresentada uma mensagem e lido o que o usuário digitar.

Tabela

Descrição gerada automaticamente

* As mensagens de leitura, então, ficarão assim ao final do procedimento já com os valores inseridos:

Interface gráfica do usuário, Texto, Aplicativo, Word

Descrição gerada automaticamente

1. Após as leituras dos valores, o programa irá executar a soma (add) entre $s1 (valor de x) e $s2 (valor de y) e salvará seu resultado em $s0.



1. No endereço 0x0040003c informa que mais uma string, a *str3*, será chamada para a tela por meio do primeiro syscall, em 0x00400048. Logo a seguir, será apresentado na tela, junto a string, o resultado da soma – observando que o código de chamada de número 1 indica que será mostrado um valor do tipo inteiro no console. Em 0x00400050, o valor em $s0 é carregado no argumento $a0 e encaminhado pelo syscall à tela. Apresentando, então, a mensagem e o resultado da soma na linha 52, finalizando o programa.

Tabela

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Exercício 2**

Implemente um laço de repetição do tipo for que conte de 0 a 9 e imprima o valor de contagem no console, conforme o exemplo abaixo:

**Em C++:**

#Disciplina: Arquitetura e Organização de Processadores

#Atividade: Avaliação 01 – Programação em Linguagem de Montagem

#Exercício 02

#Aluno: Nathalia Suzin

#include <iostream>

using namespace std;

int main() {

for(int i=0;i<10;i++)

cout << "[" << i << "]";

return 0;

}

**Em Mips:**

#Disciplina: Arquitetura e Organização de Processadores

#Atividade: Avaliação 01 – Programação em Linguagem de Montagem

#Exercício 02

#Aluno: Nathalia Suzin

#Em C++:

#include <iostream>

#using namespace std;

#int main() {

# for(int i=0;i<10;i++)

# cout << "[" << i << "]";

# return 0;

#}

.data # segmento de dados

cont: .asciiz "Contador de 1 a 10: "

.text # segmento de código

main:

addi $s0, $zero, 0 # i=0

li $v0, 4

la $a0, cont

syscall

Loop:

slti $t0, $s0, 10 # se i<10 então $t0 = 1 senão $t0 = 0

beq $t0, $zero, Exit # se $t0 = 0 então Exit

li $v0, 1

add $a0, $s0, $zero

syscall

addi $s0, $s0, 1 # $s0 + 1

j Loop

Exit: nop

▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬

1. Iniciando o programa, é incluído o valor 0 ao índice, como indicado na linha 21, endereço 0x00400000. Em seguida, a mensagem armazenada em cont é chamada à tela, informando que serão contados os valores de 0 a 9.

Tabela

Descrição gerada automaticamente

1. A partir do endereço 0x00400014 é feita uma validação em que se o valor do índice for maior que 9, então deverá ser acrescentado o valor de número 0 ao registrador, caso contrário será atribuído o valor de número 1. Logo, teremos dois casos:
2. Se o valor acrescentado a $t0 for 0, então o programa será conduzido a uma segunda validação, em 0x00400018 (linha 28), onde, se o valor em $t0 for igual a 0, fará com que ele seja deslocado para a instrução Exit, finalizando a operação do programa.
3. Se o valor acrescentado a $t0 for 1, o código seguirá no loop normalmente até a ordem de parada, ou seja, até o caso ‘a)’ acontecer.



1. Para apresentar os números que foram contados de 0 a 9, foi chamado o comando 1 (print\_int) pelo $v0, endereço 0x24020001, adicionando o valor armazenado em $s0 no argumento $a0 e introduzindo o syscall.



1. No endereço 0x00400028, observa-se que ocorre uma adição ao valor anterior, ou seja, acrescenta-se 1 ao valor de $s0, efetuando a soma dos números.



1. No console de interface, é possível visualizar a mensagem e a contagem dos valores de 0 a 9:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Já na linha 36, endereço 0x0040002c, foi aplicada a instrução j (jump). Ela conduzirá o programa à próxima instrução correspondente ao seu valor atual. Sendo assim, caso a contagem não tenha passado de 9, o jump servirá para impulsionar o código a voltar para o Loop e repetir todos as instruções anteriores novamente, prosseguindo com a contagem de valores.



1. Quando a contagem chegar em 9 e for contar o valor seguinte a ele, as validações irão parar o Loop, instruindo o código a finalizar com as operações seguindo a instrução em Exit, endereço 0x00400030.



**Exercício 3**

**Em C++:**

#Disciplina: Arquitetura e Organização de Processadores

#Atividade: Avaliação 01 – Programação em Linguagem de Montagem

#Exercício 03

#Aluno: Nathalia Suzin

#include <iostream>

using namespace std;

int main() {

int i, vet[8];

for(i=0;i<8;i++){

cout << "Entre com A[" << i << "]: ";

cin >> vet[i];

}

for(i=0;i<8;i++)

cout << "\nA[" << i << "] = [" << vet[i] << "]";

return 0;

}