

Trabalho Prático 1

Trabalho a ser desenvolvido

Implementar um programa em assembly do MIPS que terá por função a contagem da ocorrência de um determinado padrão dentro de uma lista de dados (vetorDados). Cada dado contido tanto na lista de dados, quanto na lista de dados que representam o padrão (vetorPadrao) é uma palavra (i. e. *word*, 32 bits). O vetorDados poderá conter no máximo 50 palavras, enquanto o vetorPadrao poderá conter até 5 palavras.

O programa deverá permitir a definição do número de elementos do vetor de dados, os elementos deste vetor, o número de elementos do vetor de dados que representa o padrão e os elementos deste último vetor. Como saída, deve-se informar a quantidade de padrões encontrados em vetorDados. O programa a ser implementado é descrito abaixo (a esquerda) em português estruturado, bem como o funcionamento esperado (abaixo a direita). Variações do algoritmo apresentado não serão aceitas.

```
Programa Principal() {
    // INICIALIZA TAMANHO E VETOR DE DADOS
    TamVetorDados = carregaVetor(VetorDados);

    // INICIALIZA TAMANHO E VETOR DE PADRAO
    TamVetorPadrao = carregaVetor(VetorPadrao);

    // INICIA PROCEDIMENTO DE CONTAGEM DE PADRAO
    contabilizaPadrao ← 0
    posicaoDados ← 0

    Enquanto ((posicaoDados+tamanhoVetorPadrao) <= tamanhoVetorDados){
        contabilizaPadrao ←
        contabilizaPadrao +
        encontraPadrao(VetorDados, posicaoDados, vetorPadrao, 0, TamVetorPadrao);
        posicaoDados ← posicaoDados + 1
    }

    Imprime "Quantidade de padrões contabilizados: ", contabilizaPadrao
}

int carregaVetor(int * _endereçoVetor){
    // DEFINE TAMANHO DO VETOR
    Imprime "Informe o número de dados a serem inseridos no vetor"
    Leia valor
    tamanhoVetor ← valor
    // CARREGA O VETOR DE DADOS
    posicao ← 0
    Enquanto (posicao < tamanhoVetor){
        Imprime "Informe um dado a ser inserido no vetorDados"
        Leia valor
        _endereçoVetor[posicao] ← valor
        posicao ← posicao + 1
    }

    Retorna tamanhoVetor
}

int encontraPadrao(int * _vetDados,
    int _posDados,
    int * _vetPadrao,
    int _posPadrao,
    int _tamPadrao){
    se(_vetDados[_posDados] != _vetPadrao[_posPadrao])então
        retorna 0
    senão{
        se(_posPadrao==_tamPadrao-1)então
            retorna 1
        senão{
            encontraPadrao(_vetDados,
                _posDados + 1,
                _vetPadrao,
                _posPadrao + 1,
                _tamPadrao)
        }
    }
}
```

Exemplos de funcionamento do algoritmo:

Exemplo 1:

Pressupondo que o vetor de dados (vetorDados) tenha tamanho igual a 5 e contenha os seguintes dados: 1, 2, 1, 2, 1

vetorDados					
Posição	0	1	2	3	4
Dado	1	2	1	2	1

Pressupondo agora que o vetor de dados que define o padrão (vetorPadrao) tenha tamanho igual a 2 e contenha os seguintes dados: 1, 2

vetorPadrao		
Posição	0	1
Dado	1	2

Ao executar o programa com tais entradas tem-se:

Padrão detectado nas posições (0,1) e (2,3) do vetorDados

Quantidade de padrões contabilizados: 2

vetorDados x vetorPadrao					
Posição	0	1	2	3	4
	1	2	1	2	1
	1	2	1	2	1

Exemplo 2:

Pressupondo que o vetor de dados (vetorDados) tenha tamanho igual a 5 e contenha os seguintes dados: 1, 1, 1, 1, 1

vetorDados					
Posição	0	1	2	3	4
Dado	1	1	1	1	1

Pressupondo agora que o vetor de dados que define o padrão (vetorPadrao) tenha tamanho igual a 3 e contenha os seguintes dados: 1, 1, 1

Posição	0	1	2
Dado	1	1	1

Ao executar o programa com tais entradas tem-se:

Padrão detectado nas posições (0,1,2); (1,2,3); (2,3,4) do vetorDados

Quantidade de padrões contabilizados: 3

vetorDados x vetorPadrao					
Posição	0	1	2	3	4
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1

O trabalho deve ser realizado com o auxílio do simulador MARS, que está disponível na página da disciplina.

Considerações finais

- O trabalho poderá ser realizado em trios e deve ser entregue até as 23hs e 55min do dia 10/outubro/2022 via Moodle, na sala de entregas específica para este fim.
- O trio deve colocar os seus nomes como comentário no código fonte e renomear o arquivo conforme segue:

NomeSobrenome1_NomeSobrenome2_NomeSobrenome3.asm

- O trio deve fazer o upload do arquivo (.asm) na sala de entregas do moodle até a hora estipulada. Apenas um aluno do grupo deve fazer o upload. Certifique-se que o seu trabalho está disponível na sala de entregas do moodle. Não serão aceitos trabalhos fora do prazo sob nenhuma hipótese.
- Trabalhos copiados receberão nota ZERO.
- Defina uma área de dados adequada para o programa. Acrescente variáveis, se considerar necessário.
- **Os argumentos de funções e retorno de valores de funções devem ser passados através da pilha.**

Bom trabalho