

Arquitectura de Computadores I

Recorrência

Parte Prática

21/01/2004

NOTE BEM: Leia atentamente todas as questões, comente o código usando, preferencialmente, a linguagem C e respeite a convenção de passagem de parâmetros e salvaguarda de registos. Respeite rigorosamente os aspectos estruturais e a sequência de instruções indicadas no código original fornecido, bem como as indicações sobre quais os registos a usar para cada variável.

O programa que se segue controla uma balança electrónica. Internamente possui uma tabela de preços indexada pelo código do produto. Os preços armazenados na tabela representam o custo por grama de cada produto.

Considere que a função `RegistrarProdutos` tem o seguinte protótipo:

```
float RegistrarProdutos(unsigned int* nProdutos);
```

e que o programa principal tem a seguinte codificação em C:

```
static float tabelaPrecos[] = {1.2, 5.3, 4.7, 8.9, 2.5};

void main(void)
{
    static unsigned int numProdutos;
    float custoTotal;

    while (1)
    {
        custoTotal = RegistrarProdutos( &numProdutos );

        print_str("\nNr. de produtos: ");    /* Syscall */
        print_int(numProdutos);              /* Syscall */
        print_str("\nCusto total: ");        /* Syscall */
        print_float(custoTotal);             /* Syscall */
    }
}
```

a) Defina, no segmento de dados, as constantes, as variáveis declaradas como `static`, e codifique em *assembly* do MIPS a função `main`.

b) Considere que as funções `Custo` e `LerPeso` têm os seguintes protótipos:

```
float Custo(unsigned int codProd, unsigned int peso);
unsigned int LerPeso(void);
```

Traduza para *assembly* do MIPS a função `RegistrarProdutos` cujo código em C é:

```
float RegistrarProdutos( unsigned int* nProdutos )
{
    unsigned int codProd, peso;
    float total = 0.0;

    (*nProdutos) = 0;
    codProd = 0;
    while( codProd != 99 )
    {
        print_str("Código do produto (99 para terminar):"); /* Syscall */
        codProd = read_int();
        if( codProd < 5 )
        {
            peso = LerPeso();

            total = total + Custo(codProd, peso);

            (*nProdutos)++;
        }
        else if( codProd != 99 )
        {
            print_str( "Código errado." );
        }
    }
    return total;
}
```

c) Escreva em C a função `Custo` que determina, e devolve, o custo de uma dada quantidade em gramas (`peso`) do produto cujo código é `codProd`. Traduza para *assembly* do MIPS a função `Custo`. Note que o preço por grama de cada produto é obtido por indexação da tabela `tabelaPrecos` através do código de produto.

```
float Custo(unsigned int codProd, unsigned int peso);
```

Tenha em atenção que o valor de retorno da função é do tipo `float` e deve ser devolvido através do registo `$f0`.