MC404: Organização de Computadores e Linguagem de Montagem

2ª Lista de Exercícios (24/08/2018)

Questão 1. Traduza os trechos de programa em C abaixo para linguagem de montagem do ARM.

A)

```
char  c, *pc;
int i, *pi;
...

c = 'a';
pc = &c;
*pc = 'c';

i = -1;
pi = &i;
*pi = 4096;
```

B)

```
int x=1;
char vet[100],*p;
...
p = &vet[0];
*p = (char) x; // considere dois casos: big-endian e little endian
```

Questão 2. Escreva um trecho de programa, em linguagem de montagem do ARM, que considerando que `r1` contém o endereço inicial de uma cadeia de caracteres, composta por letras na representação ASCII e espaços em branco (caractere `0x20`), terminada pelo caractere `0x00`, coloque em `r0` o número de espaços em branco que a cadeia contém.

Questão 3. Escreva um trecho de programa, em linguagem de montagem do ARM, que considerando que `r1` contém o endereço inicial de uma cadeia de caracteres, composta por letras na representação ASCII e espaços em branco(caractere `0x20`), terminada pelo caractere `0x00`, substitua todas as ocorrências de dois ou mais espaços em branco por apenas um espaço em branco. Por exemplo, se a cadeia é

```
␣␣␣␣␣␣Uma␣␣␣␣␣cadeia␣␣␣␣␣␣␣␣bem␣simples
```

seu programa deve transformá-la da cadeia

```
␣Uma␣cadeia␣bem␣simples
```

A cadeia deve ser transformada "in place", ou seja, não devem ser usadas outras regiões de memória além da que a cadeia já ocupa.

Questão 3. Escreva um trecho de programa em linguagem de montagem do ARM que implemente a soma de dois vetores $vetA$ e $vetB$ de números inteiros de 32 bits sem sinal, armazenando o resultado no vetor $vetC$. Assuma que os inícios dos vetores $vetA$, $vetB$ e $vetC$ estão respectivamente nos endereços indicados pelos rótulos `vetA`, `vetB` e `vetC`, e que o número de elementos do vetor (mesmo para todos os vetores) é dado em

uma palavra de memória armazenada no endereço de rótulo `num_elem` (ou seja, o número de elementos é o conteúdo da palavra cujo rótulo é `num_elem`).

Questão 4. Altere o trecho de programa da questão anterior para zerar todos os elementos do vetor resultado ($vetC$) caso ocorra estouro de campo na soma de algum dos elementos.