

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
2009/2010, 1.º semestre lectivo

Trabalho prático de programação em *assembly*

Este trabalho é realizado em grupo, com o máximo de três alunos. Para efeitos de avaliação, o grupo elabora um relatório sucinto, que deve incluir: código dos exercícios de programação; descrição das opções tomadas; resultados da execução.

Para cada exercício deve realizar um programa de teste e demonstração, escrito em linguagem C, que deve avaliar as condições de teste relevantes e produzir indicadores de sucesso booleanos.

1. Realize, em *assembly* da arquitectura IA-32, a função

```
int toUpperSelective(char *str, const char *selectChars);
```

que converte para maiúsculas os caracteres da *string* *str* que forem encontrados na *string* *selectChars*. A função retorna o número de caracteres substituídos.

2. Escreva, em *assembly* IA-32, o código da função

```
int getStringAddr(char *dst[], int size, const char *sample, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de apontadores para *strings* e em *dst* o endereço de um *array* de apontadores que deve preencher com os endereços das *strings* idênticas a *sample*. O parâmetro *size* é a dimensão do *array* *dst*. O fim da lista de parâmetros variável é indicado por um elemento (apontador) com o valor NULL. A função devolve o número de *strings* idênticas a *sample*.

Pode tirar partido das funções de biblioteca (por exemplo, *strcmp*).

3. Admita a seguinte declaração em C++:

```
typedef struct { int res; int average1; int average2; } results;
```

Escreva, em *assembly* da arquitectura IA-32, o código da função

```
results sumProd(const int data1[], const int data2[], int nelem);
```

que retorna uma estrutura do tipo *results* contendo: em *res* o somatório dos produtos dos elementos do *array* *data1* pelos elementos do mesmo índice em *data2*; em *average1* e *average2* as médias aritméticas dos arrays *data1* e *data2*. O número de elementos que compõem os dois *arrays* é passado no parâmetro *nelem*.

4. Escreva, em *assembly* da arquitectura IA-32, a função

```
int countIf(const void * data, int nelem, int size,  
const void * sample, int (*match)(const void *, const void *));
```

que recebe em *data* um array de dados, em *nelem* o seu número de elementos, em *size* a dimensão em *bytes* de cada elemento, em *sample* o endereço de um elemento do mesmo tipo e em *match* uma função que compara elementos desse tipo, retornando 1 se forem idênticos ou 0 se não.

A função `countIf` devolve o número de elementos idênticos a *sample* existentes no *array* *data*.

5. Escreva, em *assembly* IA-32, o código da função

```
void *getStackInfo(int *stackFrameSize);
```

que produz as seguintes informações relativas à função chamadora: endereço de retorno, indicado no valor devolvido; dimensão da *stack frame*, através da variável apontada por *stackFrameSize*.

2009.10.19

Mário Simões, Pedro Sampaio