



---

## Trabalho Prático 1

### *Introdução à Programação em Assembly*

---

**ARQUITETURA DE COMPUTADORES**

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE  
TELECOMUNICAÇÕES E COMPUTADORES

Março de 2022

## 1 Objetivos

Este trabalho tem como principais objetivos o exercício da programação em linguagem *assembly* do processador P16, incluindo a organização dos programas em rotinas e a exploração de um ambiente de programação nesta linguagem.

## 2 Especificação do Exercício

O trabalho consiste no desenvolvimento e teste de um programa para calcular valores médios da temperatura do ar, envolvendo *i)* operações com números inteiros, *ii)* manipulação de *arrays* em memória, *iii)* invocação de funções e *iv)* utilização de variáveis em memória.

Na Listagem 1 apresenta-se a especificação do programa pretendido na linguagem C, onde os tipos de dados utilizados são os definidos na biblioteca C da GNU.

O programa a desenvolver deverá ser escrito em linguagem *assembly* do P16, respeitando todas as regras da convenção P16 para a utilização de rotinas, e o seu teste deverá ser realizado recorrendo ao simulador do P16.

## 3 Trabalho a Realizar

1. Considere a definição da função `udiv` que calcula e devolve o quociente da divisão inteira de `D` por `d`, dois números inteiros codificados com 16 bits, sem sinal.

- a) Implemente a função `udiv`<sup>1</sup>.
- b) Indique, em número de bytes, a quantidade de memória de código ocupada por essa implementação. Justifique a resposta.

2. Considere a definição da função `summation` que calcula e devolve a soma dos valores dos `n` elementos de um *array*, referenciado por `a`, em que os seus elementos são números inteiros codificados com 8 bits, com sinal.

Se nos cálculos intermédios se exceder a gama representável pelo tipo `int16_t`, a função devolve o valor `INT16_MAX`. Esta constante corresponde ao maior valor possível de codificar numa variável com tipo `int16_t`. A constante `INT16_MIN` corresponde ao menor valor possível de codificar numa variável com tipo `int16_t`.

- a) Indique, justificando, os valores associados às constantes `INT16_MIN` e `INT16_MAX`.
- b) Indique duas possibilidades de implementação da constantes `INT16_MIN` e `INT16_MAX` e discuta as suas vantagens e desvantagens quanto aos requisitos de memória.
- c) Implemente a função `summation`<sup>1</sup>.

3. Considere a definição da função `average` que calcula e devolve a média dos valores dos `n` elementos de um *array*. O *array* é referenciado por `a` e os seus elementos são números inteiros codificados com 8 bits, com sinal.

- a) Indique, justificando, o registo preferível para implementar a variável `avg`: `R2` ou `R5`?
- b) Indique, justificando, o valor associado à constante `INT8_MAX`, que corresponde ao maior valor possível de codificar numa variável com tipo `int8_t`.
- c) Implemente a função `average`<sup>1</sup>.

4. Implemente a função `main` e as definições das variáveis `array1`, `array2`, `avg1` e `avg2`.

<sup>1</sup>Recomenda-se a elaboração de um programa de teste que permita verificar o comportamento da rotina desenvolvida em diversos cenários de utilização.

```
1 #define ARRAY_SIZE 12
2
3 int8_t array1[ARRAY_SIZE] = { 24, 25, 29, 34, 38, 40, 41, 41, 39, 35, 30, 26 };
4 int8_t array2[ARRAY_SIZE] = {-25, -22, -17, -5, 5, 11, 12, 9, 3, -7, -19, -24 };
5
6 int8_t avg1, avg2;
7
8 void main( void ) {
9     avg1 = average( array1, ARRAY_SIZE );
10    avg2 = average( array2, ARRAY_SIZE );
11    while( 1 )
12        ;
13 }
14
15 int8_t average( int8_t a[], uint16_t n ) {
16     int8_t avg = INT8_MAX;
17     uint16_t uacc, uavg;
18     uint8_t neg;
19
20     int16_t acc = summation( a, n );
21     if ( acc != INT16_MAX ) {
22         if( acc < 0 ) {
23             neg = 1;
24             uacc = -acc;
25         } else {
26             neg = 0;
27             uacc = acc;
28         }
29         uavg = udiv( uacc, n );
30         if( neg == 1 ) {
31             avg = -uavg;
32         } else {
33             avg = uavg;
34         }
35     }
36     return avg;
37 }
38
39 int16_t summation( int8_t a[], uint16_t n ) {
40     uint8_t error = 0;
41     int16_t acc = 0;
42
43     for( uint16_t i = 0; i < n && error == 0; i++ ) {
44         int16_t e = a[i];
45         if ( ( e < INT16_MIN - acc ) || ( e > INT16_MAX - acc ) ) {
46             error = 1;
47         } else {
48             acc = acc + e;
49         }
50     }
51     if ( error == 1 ) {
52         acc = INT16_MAX;
53     }
54     return acc;
55 }
56
57 uint16_t udiv( uint16_t D, uint16_t d ) {
58     int32_t q = D;
59     uint32_t shf_d = ((uint32_t) d) << 16;
60
61     for ( uint8_t i = 0; i < 16; i++ ) {
62         q <<= 1;
63         q -= shf_d;
64         if ( q < 0 ) {
65             q += shf_d;
66         } else {
67             q |= 1;
68         }
69     }
70     return q;
71 }
```

Listagem 1: Descrição em linguagem C do programa a desenvolver.

## 4 Avaliação

O trabalho deve ser realizado em grupo e conta para o processo de avaliação da unidade curricular.

Cada grupo deverá submeter o trabalho realizado na plataforma Moodle, na forma de listagem do programa desenvolvido, devidamente indentado e sucintamente comentado. As respostas às perguntas formuladas neste enunciado devem ser incluídas na própria listagem do programa, sob a forma de comentários.

**A data limite para a entrega dos trabalhos é 11 de abril de 2022.** Após esta entrega, o docente responsável pela lecionação das aulas teórico-práticas combinará com cada grupo uma data e hora para a realização da apresentação do trabalho.

## Bibliografia

- [1] Dias, Tiago: *Manual de consulta rápida das instruções do P16*. ISEL, Lisboa, Portugal, março 2022. [https://2122moodle.isel.pt/pluginfile.php/1148364/mod\\_label/intro/p16\\_qr\\_pt\\_v1.0.5.pdf](https://2122moodle.isel.pt/pluginfile.php/1148364/mod_label/intro/p16_qr_pt_v1.0.5.pdf) (Acedido em 14-03-2022).
- [2] Harris, Sarah e David Harris: *Digital Design and Computer Architecture: ARM Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1a edição, 2015, ISBN 978-0128000564.