

# ARQUITETURA <sup>DE</sup> COMPUTADORES

LEETC | LEIC | LEIRT



**ISEL**  
INSTITUTO SUPERIOR DE  
ENGENHARIA DE LISBOA

DEPARTAMENTO DE ENGENHARIA  
ELETRÓNICA E TELECOMUNICAÇÕES  
E DE COMPUTADORES

## 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 gerar valores pseudo-aleatórios, envolvendo *i)* operações com números inteiros, *ii)* utilização de variáveis em memória, *iii)* invocação de rotinas e *iv)* manipulação de *arrays* em memória.

Na Listagem 1 apresenta-se a especificação do programa pretendido na linguagem C, em que os tipos de dados utilizados são os definidos na biblioteca C [4].

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 poderá ser realizado recorrendo à placa SDP16 ou ao simulador do P16.

## 3 Trabalho a Realizar

1. Considere a definição da função `umul132` que realiza a multiplicação de dois números naturais codificados com 32 bits, em que o parâmetro `M` corresponde ao multiplicando e o parâmetro `m` ao multiplicador.
  - a) Implemente a função `umul132`<sup>1</sup>.
  - b) Indique, em número de bytes, a quantidade de memória de código ocupada por essa implementação. Justifique a sua resposta.
2. Considere a definição da função `srand` que afeta a variável global `seed` com o valor de uma nova *semente*.
  - a) Implemente a definição da variável `seed`, definindo as secções necessárias. Justifique a sua resposta.
  - b) Implemente a função `srand`<sup>1</sup>.
3. Considere a definição da função `rand` que implementa um gerador congruencial linear (do Inglês *Linear Congruential Generator* – LCG) para gerar números pseudo-aleatórios entre zero e `RAND_MAX` [3]. A constante `RAND_MAX` corresponde ao maior valor possível de codificar numa variável com tipo `uint32_t`.
  - a) Indique duas possibilidades de implementação da constante `RAND_MAX` e discuta as suas vantagens e desvantagens quanto aos requisitos de memória.
  - b) Implemente a função `rand`<sup>1</sup>.
4. Relativamente à definição da função `main`, indique, justificando, o registo que é preferível utilizar para implementar a variável `error`: R0 ou R5.
5. Implemente o programa apresentado na Listagem 1 usando a linguagem *assembly* do P16 e as implementações propostas nos exercícios 1, 2 e 3.

---

<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 N 5
2
3 uint16_t result[N] = { 17747, 2055, 3664, 15611, 9816 };
4
5 uint32_t seed = 1;
6
7
8 uint32_t umull32( uint32_t M, uint32_t m ) {
9
10     int64_t M_ext = M;
11     int64_t p = m;
12     uint8_t p_1 = 0;
13
14     for( uint16_t i = 0; i < 32; i++ ) {
15         if ( ( p & 0x1 ) == 0 && p_1 == 1 ) {
16             p += M_ext << 32;
17         } else if ( ( p & 0x1 ) == 1 && p_1 == 0 ) {
18             p -= M_ext << 32;
19         }
20         p_1 = p & 0x1;
21         p >>= 1;
22     }
23     return p;
24 }
25
26
27 void srand( uint32_t nseed ) {
28
29     seed = nseed;
30 }
31
32
33 uint16_t rand( void ) {
34
35     seed = ( umull32( seed, 214013 ) + 2531011 ) % RAND_MAX;
36     return ( seed >> 16 );
37 }
38
39
40 int main( void ) {
41
42     uint8_t error = 0;
43     uint16_t rand_number;
44     uint16_t i;
45
46     srand( 5423 );
47     for( i = 0; error == 0 && i < N; i++ ) {
48         rand_number = rand();
49         if( rand_number != result[i] ) {
50             error = 1;
51         }
52     }
53     return 0;
54 }
```

**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 (UC) Arquitetura de Computadores (AC).

A entrega do trabalho consiste na submissão das respostas a todas as perguntas formuladas no enunciado através da atividade "Entrega do 1.º Trabalho Prático" disponível na página de meta disciplina da unidade curricular na plataforma Moodle do ISEL, em que as listagem dos troços de código desenvolvidos devem estar devidamente indentadas e sucintamente comentadas. Recomenda-se que apenas um dos elementos de cada grupo de alunos/as faça a submissão do trabalho.

**A data limite para a entrega dos trabalhos é 31 de março de 2025.**

Após a entrega do trabalho, poderá ser combinado com algum(ns) grupo(s) uma data e hora para a realização de uma discussão para apresentação e defesa do trabalho realizado, situações que serão devidamente justificadas.

## Bibliografia

- [1] Dias, Tiago: **Manual de consulta rápida das instruções do P16**. ISEL, Lisboa, Portugal, fevereiro 2024. [https://iselpt.sharepoint.com/:b:/s/acp/EVR0vj3IkJZHp--3eH88wQUBspGUrKPOVXqGcR\\_USuoEBQ?e=Jwtpvx](https://iselpt.sharepoint.com/:b:/s/acp/EVR0vj3IkJZHp--3eH88wQUBspGUrKPOVXqGcR_USuoEBQ?e=Jwtpvx) (Acedido em 08-02-2025).
- [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.
- [3] L'Ecuyer, Pierre: **History of uniform random number generation**. Em **2017 Winter Simulation Conference (WSC)**, páginas 202–230, Las Vegas, USA, dezembro 2017.
- [4] Loosemore, Sandra, Richard M. Stallman, Roland McGrath, Andrew Oram e Ulrich Drepper: **The GNU C Library Reference Manual**, 2022. [https://www.gnu.org/software/libc/manual/html\\_node/Integers.html](https://www.gnu.org/software/libc/manual/html_node/Integers.html), acedido em 08-02-2025.