

# Projecto de Sistemas Operativos 2017-18

## heatSim

## Exercício 2

LEIC-A / LEIC-T / LETI  
IST

### Resumo

Este documento constitui um guia para a realização do segundo exercício do projecto da disciplina de Sistemas Operativos. Este exercício consiste no estudo e melhoramento da biblioteca de troca de mensagens fornecida pelo corpo docente.

## 1 Introdução

Os alunos devem ler primeiro todos os guias anteriores.

No exercício 1 foi solicitado o desenvolvimento de uma versão paralela da propagação do calor usando troca de mensagens. Para esse efeito, os alunos usaram uma biblioteca que foi fornecida pelos docentes. Neste exercício os alunos devem estudar esta biblioteca e melhorá-la.

## 2 Funcionamento da Biblioteca

A biblioteca *mplib3* mantém o estado dos canais em memória partilhada. Para sincronizar o acesso ao estado partilhado quando são trocadas mensagens, a biblioteca recorre ao suporte oferecido pelas *pthread*s para trincos (*mutex*) e variáveis de condição.

Cada canal é concretizado por uma fila duplamente ligada de mensagens. No caso em que os canais possuem capacidade superior a 0, a tarefa de enviar uma mensagem consiste em copiar os dados do emisor para um *tampão* temporário e inserir esse tampão no fim da lista do respectivo canal. A tarefa de receber uma mensagem consiste em remover o primeiro *tampão* (ou seja, mensagem) da lista do respectivo canal e copiar os dados do tampão para o receptor. A biblioteca mantém um vetor de canais, que são criados quando se invoca a primitiva de iniciação da biblioteca.

Deixa-se aos alunos a tarefa de compreender o funcionamento da biblioteca no caso em que a capacidade dos canais é igual a 0.

## 3 Melhoramento da Sincronização

Propositadamente, a sincronização do código fornecida está simplificada e possui algumas limitações.

De facto, apesar de cada canal ter a sua lista própria, a sincronização de todos os canais recorre a um único trinco e a duas variáveis de condição que são partilhadas por todos os canais.

Desta forma, se um fio de execução está a executar o envio de uma mensagem para um canal, enquanto este fio detiver o trinco todos os outros fios de execução ficam bloqueados, mesmo que pretendam ler ou

escrever noutros canais. Na prática, isto faz com que todas as operações de envio e recepção de mensagem sejam executadas em série, reduzindo a concorrência do sistema.

Pretende-se que os alunos eliminem esta limitação, desenvolvendo uma versão **que permita ler e escrever em canais distintos de forma concorrente**.

Note-se que, em teoria, é possível aumentar ainda mais a eficiência da biblioteca, permitindo que, em condições favoráveis, seja possível inserir e remover mensagens do mesmo canal de forma efetivamente concorrente. Os alunos são convidados a estudar esta solução, mesmo esta **não sendo exigida para obter a nota máxima** neste exercício.

## 4 Experimente

Uma vez que a inserção e remoção nas filas é muito rápida, com poucas tarefas pode ser difícil de observar as diferenças entre as soluções. Sugere-se que os alunos acrescentem código extra nas primitivas de envio e recepção, de forma que os fios de execução detenham o trinco por um período artificialmente longo. Este artifício deve facilitar observar as vantagens da solução que suporta mais concorrência.

## 5 Entrega e avaliação

Os alunos devem submeter um ficheiro no formato zip com o código fonte e o ficheiro *Makefile*. O exercício deve obrigatoriamente compilar e executar nos computadores dos laboratórios.

A data limite para a entrega do primeiro exercício é 25/10/2017 até às 23h59m. A submissão é feita através do Fénix.

Após a entrega, o corpo docente disponibilizará a codificação da respetiva solução, que pode ser usada pelos alunos para desenvolverem os exercícios seguintes.