

# Sistemas Operacionais – Exercício Prático

## Programação Concorrente e Sincronização

### 1 Objetivos

O objetivo desta aula prática é utilizar programação concorrente e sincronização de threads com o objetivo de solucionar um problema de maneira concorrente, bem como avaliar os aspectos positivos e negativos da programação concorrente utilizando threads.

### 2 Estruturas de Dados

O problema consiste das seguintes estruturas de dados:

- Variável inteira  $n$ ;
- Duas matrizes quadradas  $X$  e  $Y$  de dimensão  $n$ , capazes de armazenar valores inteiros.

### 3 O Problema

Todos as posições de  $Y$  devem ser iniciadas com um valor aleatório entre 1 e 10. Uma posição  $X[i][j]$  é calculada da seguinte maneira:

- Se  $i = 0$ :  $X[i][j]$  recebe um valor aleatório entre 1 e 100;
- Se  $(i > 0 \ \&\& \ j > 0 \ \&\& \ j < n - 1)$ :  $X[i][j] = \max(X[i - 1][j - 1], X[i - 1][j], X[i - 1][j + 1]) + Y[i][j]$ ;
- Se  $i > 0 \ \&\& \ (j == 0 \ || \ j == n - 1)$ :  $X[i][j] = X[i - 1][j] + Y[i][j]$ .

Utilizando a linguagem de sua preferência (Java, C/C++, Python, etc.), implemente o problema acima utilizando duas abordagens de paralelização/sincronização diferentes. Uma das abordagens deve utilizar o conceito de *barreiras*, e a outra abordagem deve utilizar o conceito de *semáforos*.

### 4 Entrega

Essa atividade pode ser feita individualmente, e a entrega acontecerá via AVA *impreterivelmente* até às 23h55 do dia (25/09).

Caso as implementações propostas estejam corretas, essa atividade acrescentará 0,5 ponto na nota da Prova 1.