

# Árvores binárias compartilhadas

Felipe Cadar Chamone  
2016006417

Setembro 2020

## 1 Objetivo

Implementar uma árvore binária que funcione em um ambiente de threads com eficiência.

## 2 Implementação

A base do programa foi feita com o código de árvores binárias fornecido pelo livro "Projeto de Algoritmos com implementações em Pascal e C" do Prof. Nívio Zivianim, como indicado na especificação do trabalho.

A solução encontrada foi aproveitar a lógica do problema de leitores/escritores usando semáforos. Para isso, cada nó da árvore deve possuir 3 semáforos: uma para controle do registro, um para o controle do contador de leitores simultâneos e um para manter a ordem das requisições.

Usando essa estrutura, montamos então 2 pares de funções, `startWrite()` e `endWrite()`, e `startRead()` e `endRead()`. Essas funções recebem como parâmetro o nó em questão e tomam conta de toda a lógica de acesso simultâneo.

A ideia era usar um `startRead()` sempre que uma thread precisar ler um nó e usar um `endRead()` quando ela acabar de ler. A mesma lógica vale para a escrita.

## 3 Desafios

O principal desafio aqui foi definir as restrições de escrita. Dado que uma thread vai alterar o nó "X", quais outros nós devem ser bloqueados para as o funcionamento correto da árvore.

Apesar de muito tempo de debug, o código ainda apresenta um comportamento estranho quando desejamos remover um nó, falhando em alguns casos que ocorre uma situação de deadlock, que por duas vezes não deveria ser possível.

## 4 Conclusão

Apesar da lógica de leitura e escrita ser bem simples, organizar as operações dentro da árvore para que um nó não interfira em outros se mostrou ser uma tarefa complexa. Creio que com mais tempo chegaríamos na solução correta, mas infelizmente ainda não é o caso.