

LABORATÓRIO 3: SEMÁFOROS

[INE5645](#) | [Descrição](#) | [Implementação](#) | [Apresentação](#)

Descrição

Nesta atividade de laboratório você deve simular uma competição de threads jogando o jogo tesoura, papel e martelo. Existem dois times de jogadores (verde e azul) com 90 threads Jogador cada - todas devem ser criadas no início do programa. Em cada rodada, três jogadores de cada time entra na região crítica e escrevem sua escolha (tesoura, martelo ou papel) num array de tamanho dois (0 para o jogador verde e 1 para o jogador azul). Existe uma thread Juiz que lê cada desses *arrays* para verificar se houve vencedor. Caso não haja, esses dois jogadores continuam jogam. Caso contrário, os dois jogadores ficam em espera do resultado dos outros jogos naquela rodada. Ao final da rodada, o Juiz verifica o resultado final daquela rodada (ex. 2x1, 3x0, 1x2 ou 0x3). Os três jogadores perdedores devem terminar sua execução e os vencedores voltam para o início da fila para entrar de novo no jogo.



O jogo termina quando um dos times não tiver mais jogadores. No fim, o programa deve imprimir a quantidade de rodadas que cada time venceu e o resultado de cada rodada - o juiz pode fazer esta contabilização.

Implementação

Crie 90 threads para cada time usando o Executor. Você deve encontrar uma forma para que entre 3 jogadores de cada time na sala de jogo (um semáforo de tamanho 6). Você deve utilizar Wait/Notify ou Await/Signal para qualquer situação em que a thread Jogador deve esperar (wait), por exemplo, esperar até que tenha 3 parceiros para entrar na região crítica ou esperar pelo resultado do juiz (saber se foi vencedor do jogo e, também, se foi vencedor da rodada). Outro caso seria um jogador escrever sua escolha no array e chamar wait (ou await) até que o juiz o acorde (notify ou signal) para saber se joga de novo ou espera (wait ou await) pela decisão da rodada.

Para que o Juiz não leia enquanto tiver jogador escrevendo no array, este deve ser implementado usando AtomicIntegerArray de duas posições.

Antes de partir para implementação do código façam a modelagem do sistema para que tenham certeza do que vocês estão fazendo. Fica mais claro para o professor ver a modelagem de vocês e depois analisar a implementação. Mostrem que vocês são bons engenheiros de software !

Apresentação

A atividade deve ser desenvolvida **em duplas**. O programa deve ser apresentado ao professor no laboratório **até o dia 04/05**. Os dois componentes do grupo devem estar presentes. Será verificado o funcionamento do programa e em seguida os alunos devem responder a questões sobre a forma como foram utilizados *threads* e semáforos no programa.

Trabalhos entregues com atraso terão desconto automático de 2 pontos. Após a segunda semana (14 dias após o fim do prazo original), o trabalho não mais será aceito, ou seja, terá nota zero.

Podem ser atribuídas notas diferentes aos alunos de um grupo, dependendo das respostas às perguntas sobre o código do programa efetuadas pelo professor. Caso um dos alunos não esteja presente ou demonstre não conhecer o código do programa, será atribuída nota zero à atividade. Em caso de **cópia** do código de outro grupo, ambos terão nota igual a **zero**.