



## Trabalho prático

### Sincronização em programas concorrentes

#### Objetivo

O objetivo deste trabalho é estimular o projeto, implementação e avaliação de soluções para problemas por meio de programação concorrente, em especial colocando em prática os conceitos e mecanismos de sincronização de processos/*threads*.

#### O Problema

Um escritório contém um banheiro que pode ser utilizado tanto por homens quanto por mulheres, mas não por ambos ao mesmo tempo. Se um homem estiver no banheiro, outros homens podem entrar, porém eventuais mulheres que desejem utilizar o banheiro devem esperar ele ficar vazio. Se uma mulher estiver no banheiro, outras mulheres podem entrar, porém eventuais homens que desejem utilizar o banheiro devem esperar ele ficar vazio. Cada pessoa (homem ou mulher) pode passar um determinado tempo utilizando o banheiro, que possui uma capacidade limite de pessoas que podem utilizá-lo ao mesmo tempo.

#### Tarefas

A tarefa central a ser realizada neste trabalho consiste em projetar e implementar uma solução concorrente para o problema anteriormente descrito utilizando conceitos e técnicas de programação concorrente, incluindo a criação, execução e sincronização de processos/*threads* independentes e concorrentes. A solução poderá ser implementada utilizando facilidades providas pelas linguagens de programação C/C++, Java **ou** Python, resultando em **dois programas distintos**, cada um utilizando um mecanismo diferente para sincronização de processos/*threads* concorrentes.

O desenvolvimento da solução deve de antemão visar pela busca de desenvolvimento de *software* de qualidade, isto é, funcionando correta e eficientemente, exaustivamente testado, bem documentado e com tratamento adequado de eventuais exceções. Mais ainda, a implementação deverá garantir correteza do programa com relação a concorrência e aplicar de forma adequada os conceitos e me-

canismos de sincronização, impedindo, portanto, a ocorrência de condições de *deadlock*, *livelock* ou *starvation* e realizando exclusão mútua de forma apropriada.

Durante sua execução, o programa deve exibir a entrada e saída de uma pessoa (homem ou mulher) do banheiro bem como quantas pessoas (homens ou mulheres) estão no banheiro no momento. Por ser um espaço de tamanho relativamente diminuto, o banheiro possui uma capacidade limite de pessoas *C* (fornecida como entrada via linha de comando ou prefixada como um valor constante) que podem utiliza-lo ao mesmo tempo e o tempo que cada pessoa passa no banheiro é randômico e diferente a cada execução do programa.

Além da implementação da solução, deverá ser elaborado um relatório escrito simples descrevendo, pelo menos:

- como a solução foi projetada;
- a lógica de sincronização utilizada, em termos dos mecanismos empregados e como ela é feita entre os fluxos de execução do programa;
- como é garantida a corretude da solução com relação a concorrência;
- uma análise comparativa entre os diferentes mecanismos de sincronização utilizados, em particular com relação a esforço de desenvolvimento e complexidade;
- eventuais dificuldades encontradas durante o desenvolvimento, e;
- instruções para compilação e execução do programa.

## Autoria e política de colaboração

O trabalho poderá ser feito **individualmente ou em equipe composta por no máximo dois estudantes**, sendo que, neste último caso, é importante, dentro do possível, dividir as tarefas igualmente entre os integrantes da equipe. O trabalho em cooperação entre estudantes da turma é estimulado, sendo aceitável a discussão de ideias e estratégias. Contudo, tal interação não deve ser entendida como permissão para utilização de (parte de) código fonte de outras equipes, o que pode caracterizar situação de plágio. Trabalhos copiados em todo ou em parte de outras equipes ou da Internet serão sumariamente rejeitados e receberão nota zero.

A critério do professor, qualquer equipe pode ser eventualmente convocada para uma entrevista cujo objetivo é confirmar a autoria do trabalho desenvolvido e determinar a contribuição real de cada integrante. Durante a entrevista, cada membro da equipe deverá ser capaz de explicar, com desenvoltura, qualquer parte do trabalho, mesmo que esta tenha sido desenvolvida por outro membro da equipe. Portanto, é possível que ocorra, após a entrevista, redução da nota geral do trabalho ou ajustes nas notas individuais com vistas a refletir a verdadeira contribuição de cada membro da equipe.

## Entrega

Você deverá submeter um único arquivo compactado no formato .zip contendo todos os códigos fonte resultantes da implementação deste trabalho, sem erros de compilação e devidamente testados e documentados, **até as 23h59 do dia 1º de novembro de 2017** através da opção *Tarefas* na Turma Virtual do SIGAA. Juntamente com os códigos fonte, o arquivo compactado deverá também conter o relatório escrito, preferencialmente em formato PDF. Se for o caso, é possível fornecer, no campo *Comentários* do formulário eletrônico de submissão da tarefa, o endereço de um repositório remoto destinado ao controle de versões, porém esta opção não exclui a necessidade de submissão dos arquivos via SIGAA.

## Avaliação

A avaliação deste trabalho será feita principalmente sobre os seguintes critérios: (i) utilização correta dos conceitos e mecanismos de sincronização de processos/*threads*; (ii) a corretude da execução dos programas implementados, tanto com relação a funcionalidades quanto a concorrência; (iii) a aplicação de boas práticas de programação, incluindo legibilidade, organização e documentação de código fonte, e; (iv) qualidade do relatório produzido. O trabalho possuirá nota máxima de 6,0 (seis) pontos, nota essa que será contabilizada para a segunda unidade da disciplina.