



Trabalho Prático: Mecanismos de Sincronização

Frank Laércio da C. S. Júnior e Ohanna de Macena Dezidério

frank.junior.118@ufrn.edu.br e ohanna.deziderio.121@ufrn.edu.br

Solução

O algoritmo se baseou no problema do produtor e consumidor apresentado em sala de aula durante a disciplina de Programação Concorrente. Logo o *buffer* comum compartilhado é o banheiro, a capacidade do *buffer* é quantas pessoas podem acessar simultaneamente o banheiro, os produtores são as pessoas que estão querendo usar o banheiro e os consumidores são as pessoas que estão saindo do banheiro. Além disso, para gerar os gêneros e o tempo de uso do banheiro foi utilizada a função *random* do Java apenas utilizando a implementação padrão “(*new Random()*)”.

Sincronização

No projeto foram utilizados dois mecanismos de sincronização: o monitor e o *notify*. O monitor foi utilizado nos métodos ``use()`` e ``exit()``, pois é onde pode-se entrar e sair do banheiro. Logo, deve-se ser feita uma sincronização no *buffer* compartilhado para que o produtor não produza demais e nem consumidor fique sem consumir, evitando assim o problema de *starvation*. O *notify* foi implementado para notificar a outros produtores que estejam aguardando usar o *buffer* que podem entrar no banheiro. Além disso, ele foi utilizado quando o banheiro estava ocupado com o método “*wait()*”, para aguardar que o banheiro esteja livre para uso.

Corretude

A corretude do programa é garantida pelos mecanismos de controle para saber se o *buffer* está vazio ou ocupado. Caso o *buffer* esteja ocupado é necessário aguardar o

buffer ter espaço, liberado no método “*exit()*” e se o *buffer* estiver vazio a *thread* fica aguardando os consumidores.

Dificuldades

As dificuldades foram principalmente enfrentadas no mecanismo de sincronização, pois estava ocorrendo *deadlock* e na implementação do tempo de uso do banheiro, pois tivemos que adaptar a solução com uma nova classe “*Person()*” que tem os atributos de gênero e tempo de uso do banheiro.

Instruções de como executar

1. Ter o Java 17 ou posterior instalado na máquina.
2. Fazer o download do código-fonte.
3. Executar o projeto Java.
4. Caso queira alterar a quantidade da capacidade ou número de pessoas, deve-se ajustar a classe **Main** do projeto.