

Relatório Mini-Projeto 2

Server

Turma 6 - Grupo 3
Adelaide Santos
Bruno Mendes
Eunice Amorim
Rita Mendes

up201907487
up201906166
up201904920
up201907877

Main thread - função listener

O Cliente envia continuamente pedidos ao Servidor durante o seu tempo de execução. Deste modo, implementou-se a função *listener* que lê os pedidos da FIFO pública e que lança um novo produtor por cada um deles. Contudo, ela não pode deixar de fazer essa leitura a quando de *timeout* por parte do Servidor. Enquanto o Cliente se encontrar ativo, é necessário continuar a receber pedidos para lhes responder que o serviço encerrou. Como tal, o ciclo de leitura é parado se o tempo de execução do Servidor se tiver esgotado e, ao mesmo tempo, já não for possível ler uma mensagem.

Criação das *threads* produtoras e consumidora

A *thread* consumidora, responsável por responder aos pedidos do Cliente é única, como tal, a sua gestão não gera problemas. Com tudo, o número de *threads* produtora, existindo uma por cada pedido, tende para infinito dadas as condições de competição. Desta forma, não existe uma estrutura grande o suficiente para guardar todas as informações das *threads* a dar *join* no final. Para tal, optou-se pelo uso de *threads detached* para os produtores.

Implementação de filas

Os pedidos devem ser atendidos por ordem de chegada, ou seja, seguindo um protocolo *First In, First Out*. Deste modo, a estrutura ideal para armazenar os dados de pedidos seria uma fila. Por este motivo, decidiu-se implementar essa estrutura. Utilizaram-se assim, duas filas. Uma para colocar os pedidos a serem processados e outra com os finalizados prontos para serem passados aos clientes pelo consumidor. Tal implica uma gestão recorrendo a *mutexes* para evitar operações de *push* e *pop* concorrentes e de *condition variables* para evitar que o tamanho ultrapasse o máximo estipulado. Assume-se que o tamanho passado com argumento é o tamanho máximo que cada uma das filas pode assumir.

Registo de operações

O registo de operações é feito com base no esquema pedido através da escrita em `stdout` com *printf*.

Participação de cada elemento

O esforço de trabalho foi dividido por todos de igual forma.

#teamWork