Relatório Sistemas Operativos

Gonçalo Leal Gonçalves Geraldes – 2021133456

Pedro Miguel Dinis Sequeira – 2020132079

## **Arquitetura geral**

## 

## O named pipe BACKENDFIFO é criado pelo programa backend e o FRONTENDFIFO pelo programa frontend; o pipe p\_b é criado pelo backend e usado num fork para correr o programa do respetivo promotor.

## **Header file header.h**

Contém as estruturas de dados cliente e leilao, que armazenam os dados dos clientes que estão a usar o sistema e itens à venda no leilão; existem ainda estruturas de dados responsáveis pela comunicação entre o backend e o frontend, que armazenam o comando do utilizador para que seja passado ao backend (struct msgBF) e a resposta do mesmo (struct respostaBF).

Contém ainda os includes das bibliotecas standard e defines dos named pipes usados na comunicação backend frontend.

## **Frontend.c**

O frontend começa por validar se o utilizador introduziu o seu nome e password, acabando o programa em caso contrário.

Quando um utilizador corre o frontend (corretamente) entra em ciclo onde pode inserir comandos. Estes comandos são passados ao backend através no named pipe BACKENDFIFO e é recebida uma resposta através do named pipe FRONTENDFIFO criado pelo próprio frontend, o nome de cada FRONTENDFIFO é CLIENTE%d sendo %d o pid de cada cliente, deste modo se existirem vários clientes a resposta do backend é enviada pelo respetivo pipe do respetivo cliente.

O frontend contém ainda a thread heartBeat responsável pela execução dessa mesma funcionalidade, se um cliente encerrar indevidamente a plataforma a thread informa o backend, que encerra o devido frontend.

## **Backend.c**

No backend o administrador tem a opção de inserir comandos, todos os comandos se encontram funcionais.

O backend é responsável pela gestão geral do leilão, através da thread decreaseTime o backend sabe quando os itens expiram sendo ou não vendidos a utilizadores que tenham feito uma licitação; a thread check\_active\_users verifica que utilizadores ficaram inativos e remove-os da plataforma; e a thread imprimePromotor é responsável pelo contínuo lançamento de novos promotores.