Sistemas Operativos (EIC0027)

Turma 2 Grupo 9

Filipe Carlos de Almeida Duarte da Cunha Nogueira - 201604129
Gustavo Speranzini Tosi Tavares - 201700129
Nuno Rodrigues de Castro Santos Silva - 201404676

16 de Maio de 2019

Mestrado Integrado em Engenharia Informática e Computação

Comunicação Cliente - Servidor / Servidor - Cliente:

Para a troca de mensagens entre o cliente (**user.c**) e o servidor (**server.c**), são utilizados FIFOs, em que o cliente preenche uma struct de formato *tlv_request_t* de acordo com a operação desejada nos seus argumentos de entrada, envia para o **server** através do fifo de nome **/tmp/secure_srv** , e após autenticação e validação do pedido, preenche uma struct de formato *tlv_reply_t* para responder ao user pelo FIFO **/tmp/secure_XXXXX**, em que **XXXXX** equivale ao **PID** do **user**.

Após processar o pedido, a struct **reply** retorna um dos seguintes códigos de retorno, para informar a ocorrência de algum erro ou o sucesso da operação.

RC_OK	*RC_SRV_DOWN*
RC_SRV_TIMEOUT	*RC_USR_DOWN*
RC_LOGIN_FAIL	*RC_OP_NALLOW*
RC_ID_IN_USE	*RC_ID_NOT_FOUND*
RC_SAME_ID	*RC_NO_FUNDS*
RC_TOO_HIGH	*RC_OTHER*

Mecanismos de Sincronização:

Mutexes:

De maneira a fazer a sincronização do acesso das Threads às contas bancárias, é utilizado o mutex *account_mutex* em cada conta bancária criada, de modo a assegurar que apenas uma Thread pode aceder a uma conta específica.

Assim que uma Thread acaba o processo de Autenticação do ID e password introduzida por um utilizador e está pronta a realizar o seu pedido, esta tenta bloquear o mutex da conta bancária em questão, apenas desbloqueando quando acabar de realizar o seu pedido.

Isto assegura que apenas uma thread de cada vez pode aceder e editar a informação de uma conta bancária.

No caso da operação de transferência, a Thread que tiver a conta remetente tentará bloquear o mutex da conta destinatária.

Durante o encerramento do servidor, é utilizado o mutex *server_shutdown_mutex* para bloquear a variável *server_shutdown*, de maneira a não poder ser alterada por outras Threads.

Semáforos:

De forma a fazer a sincronização do acesso das threads á fila de pedidos pendentes a serem tratados, é utilizado um semáforo que apenas permite a uma thread aceder à fila de pedidos de cada vez.

Assim que uma Thread livre tenta aceder à fila, e esta não estiver bloqueada por outra thread, a thread livre bloqueia o acesso e procede a ler e remover o próximo pedido a ser tratado. Assim que acaba a leitura, a Thread desbloqueia o acesso à Fila.

Caso não hajam pedidos na fila, a thread livre que tenha bloqueado fila começa um loop de verificação á espera que caiam mais pedidos, apenas desbloqueando quando receber um pedido a ser tratado.

Desta maneira garantimos que apenas uma Thread livre acede à fila de cada vez, não permitindo o overload de Threads “em loop” a verificar se existem pedidos novos na fila, poupando recursos do utilizador.

Encerramento do Servidor :

O Encerramento do servidor ocorre através da operação shutdown, que só pode ser efetuada pelo administrador. Após autenticação e validação do pedido, o programa fecha as threads responsáveis pelos balcões eletrônicos e posteriormente destrói o fifo responsável pelo recebimento de pedidos e o mutex.