



Universidade do Porto

FEUP Faculdade de Engenharia

MIEIC – Sistemas Operativos

Trabalho Prático II - Simulação de uma máquina multibanco

O objectivo deste projecto era criar uma aplicação que simule uma máquina multibanco, utilizando FIFOs para a comunicação entre os seus programas (Cliente, Admin e Servidor).

Estruturas utilizadas para a comunicação entre os programas

São utilizadas duas *structs*, Pedido e Resposta. A *struct* Pedido é utilizada pelo Cliente e pelo Admin para comunicar com o Servidor através do FIFO 'requests' e é constituída por:

- pid_t - pid do Cliente, para o Servidor saber o nome do FIFO para onde deve enviar a resposta;
- char[] - o nome do cliente;
- inteiro com o numero da conta origem;
- inteiro com o PIN inserido;
- inteiro com o numero da conta destino, utilizado quando a operação é uma transferência entre duas contas;
- double com a quantia em euros, utilizado quando se deposita ou levanta dinheiro e nas transferências;
- inteiro com o tipo de operação, que indica ao Servidor qual é a operação para que este saiba quais os elementos necessários

A struct Resposta é enviada pelo Servidor para o Admin e para o Cliente através de um FIFO criado anteriormente pelo Admin/Cliente, e contém:

- um inteiro com o estado, que indica ao Servidor o sucesso da operação pretendida;
- um array de chars com a mensagem que deve ser mostrada ao utilizador;

Há um caso especial nesta comunicação. Quando o Admin pede a listagem das contas de clientes existentes no Servidor. É utilizada uma variável para indicar ao Servidor, que em vez de ser enviada uma *struct* do tipo Resposta será enviado *structs* do tipo Conta, que contêm as informações da conta de um cliente, até ter enviado todas as que estão ativas no Servidor.

Quando se inicia o Servidor, este cria o FIFO 'requests' e fica à espera de pedidos de clientes ou do Admin, que depois de enviarem o pedido esperam pela Resposta no FIFO que criaram para o efeito.

Mecanismos de sincronização das threads

As contas dos clientes estão organizadas num array de apontadores para Contas no Servidor que é utilizado nas diversas funções do Servidor. Cada pedido é processado através de uma thread e, de forma a controlar o acesso ao array de Contas e prevenir que seja modificado ao mesmo tempo por mais que uma thread, é utilizado um mutex. Este mutex é bloqueado antes de uma função modificar o array de Contas. Só é desbloqueado quando a função que o bloqueou já não pretender realizar mais nenhuma alteração no array das Contas. Depois de desbloqueado já permite o acesso de outra thread ao array de Contas.

Encerramento do Servidor

Para encerrar o Servidor, é utilizada a função *pthread_exit*, que assegura que os pedidos que o Servidor recebeu antes de receber a ordem para encerrar, sejam atendidos. É chamada quando o Admin pede o encerramento do Servidor, ou quando sejam utilizadas ferramentas do sistema como kill ou ctrl-C (nestes últimos casos utilizando um *signal handler*). Além disso, as informações dos clientes são guardadas no ficheiro sempre que uma função as modifica, de forma a evitar a perda de informação caso haja um encerramento prematuro.

Miguel Oliveira | ei10076@fe.up.pt

Afonso Caldas | ei10051@fe.up.pt

Turma 5 – Grupo 7 – 25 de Maio de 2012