

# Enunciado do Projecto de Sistemas Operativos 2016-17

## Serviço de banca paralelo - Exercício 4

LEIC-A / LEIC-T / LETI  
IST

### 1 Registo de operações, redireção de saídas e acesso remoto no i-banco

O quarto exercício pretende estender o i-banco com algumas funcionalidades que facilitam a sua utilização por diferentes operadores. As subseções seguintes descrevem cada uma dessas funcionalidades.

O desenvolvimento da solução deste exercício deve assentar no código que entregaram anteriormente (ou solução publicada pelo corpo docente caso optem por isso), sem a modificação da alínea extra realizada durante a aula de avaliação.

#### 1.1 Registo da operações realizadas

Para efeitos de monitorização do sistema por uma entidade supervisora, pretende-se que todos os comandos realizados pelas tarefas trabalhadoras sejam registados num diário de operações persistente.

Mais precisamente, o i-banco deve passar a registar num ficheiro cada comando que é executado e terminado por cada tarefa trabalhadora.

Esse registo deve ser feito num ficheiro chamado `log.txt` criado na mesma diretoria onde o i-banco se executa. A cada comando terminado, a tarefa trabalhadora que o executou deve escrever uma nova linha no final do ficheiro segundo o formato `TID: comando`, em que `TID` denota o identificador da tarefa e `comando` denota o comando e respetivos argumentos.

A ordem dos comandos guardados no diário de operações (`log`) deve respeitar a ordem pela qual os comandos foram efetivamente executados. Em particular, qualquer par de comandos envolvendo a mesma conta devem surgir no diário pela ordem pela qual foram efetivamente executados.

#### 1.2 Redireção do *output* das simulações

Pretende-se também que o *standard output* (`stdout`) das simulações (processos filho lançados pelo i-banco) passe a ser redirecionado para ficheiros individuais na diretoria onde o i-banco foi lançado.

Cada processo filho lançado deve ter o seu *standard output* redirecionado para um ficheiro chamado `i-banco-sim-PID.txt`, criado na mesma diretoria em que corre o i-banco, em que

PID se refere ao pid do processo filho. No caso desse ficheiro já existir antes, o ficheiro anterior deve ser eliminado.

### 1.3 *Remote banking*

Para aumentar a capacidade de utilização da máquina onde corre o **i-banco**, pretende-se permitir que diferentes utilizadores na mesma máquina introduzam comandos no **i-banco** através de diferentes terminais.

Para tal, o **i-banco** deixa de ler comandos diretamente a partir do *standard input (stdin)*. Em alternativa, o **i-banco** passa a ter um *pipe* com nome por onde recebe comandos. O *pipe* deve ter o nome **i-banco-pipe** e estar localizado na diretoria onde o **i-banco** foi lançado, ou na diretoria **/tmp/i-banco-pipe** caso existam problemas com privilégios de acesso para escrita.

Devem ser implementados os seguintes pontos:

1. Deve ser criado um programa **i-banco-terminal** que, correndo num processo autónomo, lê ordens do teclado e envia os comandos já interpretados para o *named pipe* alvo.

O envio é feito sob a forma de um sequência de bytes correspondente ao conteúdo (binário) de uma estrutura do tipo **comando.t**. Deve assumir-se que tanto os processos terminais como o processo **i-banco** representam a estrutura **comando.t** de forma idêntica em memória.

A interpretação dos comandos deixa de ser feita no **i-banco** e passa para o programa **i-banco-terminal**. O *pathname* do *named pipe* é indicado como argumento na linha de comando do **i-banco-terminal**. Em cada momento poderá haver múltiplos processos a correr este programa e a enviar concorrentemente comandos para o mesmo **i-banco**.

2. O programa **i-banco-terminal** deve aceitar o comando *sair-terminal* que permite sair do programa **i-banco-terminal** sem terminar o **i-banco**. Nota: o comando **sair** mantém o mesmo comportamento, terminando o programa **i-banco**.
3. O programa **i-banco-terminal** deve mostrar no ecrã o *output* dos comandos introduzidos pelo utilizador e executados no **i-banco**. O formato é o mesmo dos trabalhos anteriores (1 linha de texto com o resultado da operação). Deve mostrar também, numa nova linha, o tempo total de execução do comando no **i-banco**, medido na aplicação **i-banco-terminal**.  
O **i-banco** não deve mostrar o *output* das operações. Nota: Caso não consigam implementar esta funcionalidade na totalidade, os resultados podem ser mostrados no **i-banco**, contando parcialmente para a avaliação.
4. Enquanto o *output* não for impresso, o terminal não avança para ler o próximo comando do teclado.

## 2 Experimente

Para verificar a implementação dos requisitos siga o seguinte procedimento:

1. Abrir 4 terminais.
2. No terminal 1 lançar o comando `i-banco`.
3. No terminal 2 verificar que foi criado o ficheiro `i-banco-pipe`.
4. No terminal 3 lançar o comando `i-banco-terminal i-banco-pipe`, ou `i-banco-terminal /tmp/i-banco-pipe`, e escrever `simular 4`.
5. No terminal 2 verificar que foi criado um ficheiro `i-banco-sim-PID.txt` em que PID é o identificador do processo (filho) que foi criado pelo processo `i-banco` na execução do comando `simular`. Verificar que o conteúdo desse ficheiro é o output do comando `simular`.
6. Verificar se o ficheiro `log.txt` é criado, e se nele é escrito o conteúdo correspondente aos comandos executados pelo `i-banco`.
7. Em seguida no terminal 4 lançar o comando `i-banco-terminal i-banco-pipe` e escrever `simular 5`.
8. Voltar a verificar o ponto 5.
9. Finalmente, verificar que por cada comando `simular` que introduzir no terminal 3 ou 4 será criado o correspondente ficheiro `i-banco-sim-PID.txt`.
10. Escrever `sair-terminal` num dos terminais que executa `i-banco-terminal`. Verificar que esse processo `i-banco-terminal` termina, mas os demais continuam em execução.

### 3 Funções do Unix/Linux

Para este exercício sugerimos que sejam usadas as seguintes funções da API do Unix/Linux: `unlink`, `mkfifo`, `open`, `close`, `dup/dup2`, `read`, `write` e `difftime`.

### 4 Entrega e avaliação

Os alunos devem submeter um ficheiro no formato zip com o código fonte e `makefile` através do sistema Fénix. O exercício deve obrigatoriamente compilar e executar nos computadores dos laboratórios.

A data limite para a entrega do primeiro exercício é 2 de dezembro até às 23h59m.

### 5 Cooperação entre Grupos

Os alunos são livres de discutir com outros colegas soluções alternativas para o exercício. No entanto, *em caso algum* os alunos podem copiar ou deixar copiar o código do exercício. Caso duas soluções sejam cópias, ambos os grupos reprovarão à disciplina.