Enunciado do Projeto de Sistemas Operativos 2016-17
Serviço de banca paralelo - Exercício 3

LEIC-A / LEIC-T / LETI
IST

1 Transferências

Como primeiro objetivo do terceiro exercício pretende-se introduzir uma nova operação no `i-banco`: transferência entre contas bancárias. Esta operação segue a seguinte sintaxe: `transferir idContaOrigem idContaDestino valor`.

Esta operação subtrai o valor indicado à conta de origem (caso esta tenha saldo suficiente) e credita-o na conta de destino.

1.1 Detalhes de implementação

A implementação desta operação deve maximizar o paralelismo. No entanto, não deve ser permitido o acesso a nenhuma das contas bancárias envolvidas durante a execução da operação. Deve dedicar-se especial cuidado à possibilidade de situações de interblocagem. É também necessária a inclusão de uma novo campo (que representa a conta de destino) na estrutura `comando_t` apresentada no enunciado anterior.

Em caso de sucesso e em caso de erro devem ser apresentadas mensagens no *stdout* com as seguintes estruturas, respetivamente:

- `transferir(idContaOrigem, idContaDestino, valor): OK`
- `Erro ao transferir valor da conta idContaOrigem para a conta idContaDestino.`

Sendo que `idContaOrigem`, `idContaDestino` e `valor` são os valores das variáveis recebidas como input da operação.

Sugere-se a consulta das funções: *pthread_mutex_init*, *pthread_mutex_destroy*, *pthread_mutex_lock* e *pthread_mutex_unlock*.

2 Sincronização de simulações

O segundo objetivo do terceiro exercício é a sincronização do lançamento de simulações com as tarefas trabalhadoras. Quando a tarefa principal recebe o comando `simular`, deve aguardar que todos os comandos à espera de serem executados ou em exececução pelas tarefas trabalhadoras sejam completados, antes de proceder à criação do processo filho que executará a simulação.

Caso não houvesse esta sincronização, a simulação poderia deparar-se com trincos fechados (por tarefas trabalhadoras em execução no momento do *fork*) no acesso às contas bancárias, pondo em risco a capacidade da simulação fazer progresso.

2.1 Detalhes de implementação

A implementação da solução deve recorrer a variáveis de condição, usando as funções da API do Unix/Linux estudadas na teoria para o efeito. Sugere-se a consulta das funções: *pthread_cond_init*, *pthread_cond_destroy*, *pthread_cond_wait* e *pthread_cond_signal*.

3 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 terceiro exercício é 11/11/2016 até às 23h59m. A submissão é feita através do Fénix.

Após a entrega, o corpo docente disponibilizará a codificação da respetiva solução, que pode ser usada pelos alunos para desenvolverem o exercício final.

4 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.