

Sistemas Operacionais

Gerenciamento de Processos

✔ Projeto 1

- Pontuação máxima 3,0
- Grupo de até 5 alunos no máximo
- Data da entrega: 26/09
- Definição:
 - Crie um código em C para manipular uma transação bancária entre dois clientes (duas contas, cada cliente é uma thread)
 - Cada cliente irá transferir dinheiro para o outro simultaneamente, ou seja, o cliente A transfere para o cliente B e o cliente B transfere para o cliente A, para efetuar a transferência, um cliente precisa acessar o seu arquivo com saldo (leitura) e o arquivo do outro cliente com o saldo (gravação).
 - O saldo dos clientes deve ser manipulado em 2 arquivos .txt, um chamado saldoA.txt e outro chamado saldoB.txt
 - **Teste os seguintes cenários:**
 1. Implemente uma versão do código usando somente thread, sem usar qualquer método de sincronismo, execute pelo menos 3 vezes e elabore um relatório do resultado obtido.
 2. Utilize a técnica de semáforo para travar a região crítica, ou seja, você deve travar de alguma forma a conta do cliente A e B em cada transação, pode usar algum recurso já disponível na linguagem. Execute pelo menos 3 vezes e elabore um relatório do resultado obtido. (ordem óbvia: **cliente A trava o recurso saldoA.txt e depois o recurso saldoB.txt**, por sua vez, o **cliente B trava o recurso saldoB.txt e depois o recurso saldoA.txt**)
 3. Como a tendência é travar a execução no cenário 2, proponha uma solução para evitar o travamento. Execute pelo menos 3 vezes e elabore um relatório do resultado obtido.

- <https://www.tutorialspoint.com/deadlock-with-mutex-locks>
- <http://m.blog.chinaunix.net/uid-14735472-id-4929625.html>
- https://linux.die.net/man/3/pthread_create
- https://linux.die.net/man/3/pthread_join
- https://pubs.opengroup.org/onlinepubs/007908799/xsh/pthread_mutex_init.html
- https://linux.die.net/man/3/pthread_mutex_lock
- https://linux.die.net/man/3/pthread_mutex_unlock