

Questão 1 – Saque e Depósito

Há uma variável compartilhada chamada SALDO, iniciando em 100. Duas “tarefas” (processos) acessam o mesmo recurso:

- **T_SAQUE realiza 100 saques de 1 unidade.**
- **T_DEPOSITO realiza 100 depósitos de 1 unidade.**

Tarefas:

- 1. Mostre, em pseudocódigo, como ocorre a execução sem sincronização, evidenciando o problema de condição de corrida.**
- 2. Implemente a versão correta, utilizando o algoritmo de espera ocupada via software para garantir exclusão mútua ao atualizar o saldo.**
- 3. Indique qual deve ser o valor correto de SALDO ao término da execução e explique o motivo.**

Restrições:

- **Não utilizar semáforos, mutex, primitivas atômicas ou comandos de bloqueio.**
- **Utilizar apenas flags, variável TURN e laços ENQUANTO para a sincronização.**

Questão 2 – Gravação e Leitura em Arquivo (produtor/consumidor com espera ocupada)

Um processo ESCRITOR produz linhas de texto em um BUFFER e, ao completar um bloco de N itens, grava o conteúdo no arquivo DADOS.TXT.

Outro processo, o LEITOR, deve aguardar até que o escritor sinalize que o bloco está pronto, e então realizar a leitura do arquivo.

Tarefas:

- 1. Implemente o processo ESCRITOR, garantindo que o acesso ao BUFFER seja protegido pelo algoritmo de Exclusão Mútua (escritor = ID 0, leitor = ID 1).**
- 2. Implemente o processo LEITOR, que deve aguardar até que o escritor sinalize PRONTO = VERDADEIRO antes de realizar a leitura.**
- 3. Utilize flags de estado e espera ocupada (ENQUANTO) para coordenar os estados PRONTO e LIDO, garantindo que o leitor nunca leia um bloco parcial.**