

Lista de exercícios 4

Programação Concorrente e distribuídas

Escolha 2 atividades para serem feitas e enviadas

1. Contador Multithreaded:

Escreva um programa em Python que utilize threads para contar de 1 até um número N fornecido como entrada. Cada thread deve ser responsável por incrementar o contador em uma unidade. Utilize semáforos para garantir que cada thread atualize o contador de forma exclusiva. Ao final da contagem, o programa deve imprimir o valor final do contador.

2. Produtor-Consumidor com Buffer Limitado:

Implemente um programa em Python que simule um problema de produtor-consumidor com um buffer limitado. Crie duas threads, uma produtora e outra consumidora. O produtor deve gerar itens e colocá-los em um buffer compartilhado, enquanto o consumidor deve retirar itens desse buffer. Utilize semáforos para controlar o acesso ao buffer e garantir que ele não ultrapasse sua capacidade máxima.

3. Barbearia com Múltiplos Barbeiros:

Modele o problema da barbearia com múltiplos barbeiros utilizando threads em Python. Tenha um número fixo de cadeiras na barbearia e um número variável de barbeiros. Os clientes chegam à barbearia e verificam se há cadeiras disponíveis. Se todas as cadeiras estiverem ocupadas, eles saem; caso contrário, eles ocupam uma cadeira e aguardam serem chamados por um barbeiro. Os barbeiros, quando disponíveis, chamam o próximo cliente e o atendem. Utilize semáforos para controlar o acesso às cadeiras e garantir a exclusividade do atendimento.

4. Leitores e Escritores:

Desenvolva um programa em Python que simule o problema dos leitores e escritores. Tenha uma variável compartilhada que represente um recurso que pode ser lido e escrito. Vários leitores podem ler o recurso simultaneamente, mas apenas um escritor pode escrever nele por vez. Utilize semáforos para controlar o acesso ao recurso e garantir que não ocorra condição de corrida entre leitores e escritores.

5. Jantar dos Filósofos:

Implemente uma solução em Python para o problema do jantar dos filósofos. Tenha N filósofos sentados em uma mesa, onde cada filósofo precisa de dois garfos para comer. Os garfos são compartilhados entre os filósofos, sendo que cada garfo é compartilhado por dois filósofos adjacentes. Utilize threads para representar os filósofos e semáforos para representar os garfos. Garanta que os filósofos não entrem em deadlock ao tentar pegar os garfos.