

Nome : Felipe Marques da Silva de Almeida.

Questões: Jantar dos Filósofos

1. Existe alguma maneira de implementar o jantar dos filósofos sem o uso de semáforos? Se sim, como seria? (OBS: pode ser necessário aprofundamento de pesquisa, cite e explique possibilidades, sem se preocupar (ainda) com a implementação em Java).

R. Até onde sei, é possível representar o problema do jantar dos filósofos sem o uso de semáforos. A alternativa é usar um modelo de comunicação baseado em mensagens, onde cada filósofo é representado como uma thread - ou processo - independente e se comunica com os outros filósofos por meio da troca de mensagens. Inclusive, na disciplina de Escalabilidade Combinatório, o professor João fez uma representação inicial utilizando Python desse modelo, se não me engano.

2. E se quiséssemos introduzir um garçom que supervisionasse a mesa e evitasse que mais de dois filósofos tentassem comer ao mesmo tempo, como isso poderia ser feito?

R. Acredito que sim. Pois um “garçom”, a meu ver, ajudaria a evitar problemas relacionados a “deadlock” no jantar. O “garçom” seria um “administrador central” que daria a permissão.

3. Suponha que em vez de 5 filósofos, tivéssemos um número arbitrário. Como você ajustaria o programa para lidar com isso de forma eficiente?

R. Acredito que seria necessário modificar a aplicação para acomodar dinamicamente a quantidade de “filósofos” e possibilitar que o “administrador central”, o garçom, controle o número de “filósofos” que podem comer.

4. É possível que um filósofo coma mais de uma vez antes dos outros começarem a comer? Explique.

R. Sim, é possível que um filósofo coma várias vezes antes de outros filósofos começarem a comer, e isso está sujeito à implementação e ao agendamento do sistema. A execução de threads em um programa multithread é gerenciada pelo S.O e sua ordem de execução depende de como o sistema decide agendar as threads.

5. Como você poderia modificar o programa para que cada filósofo tivesse um tempo de pensamento e de comer diferente?

R. Para dar a cada filósofo tempos de pensamento e de comer diferentes, é necessário introduzir durações aleatórias usando uma função de geração de números aleatórios.

6. E se quiséssemos que os filósofos tivessem diferentes prioridades ao comer? Como isso poderia ser implementado?

R. Isso pode ser feito definindo prioridades específicas para cada thread filósofo usando a funcionalidade de prioridade de threads oferecida pela linguagem.

7. Como você poderia usar o monitoramento de recursos para detectar problemas de deadlock no programa? (Pesquisa adicional)

R. O monitoramento de recursos pode ser usado para detectar problemas de deadlock no programa implementando um mecanismo que rastreia o estado dos recursos (garfos, no caso do problema do jantar dos filósofos) e as solicitações de aquisição e liberação desses recursos pelas threads (filósofos). Aqui estão os passos gerais para fazer isso: