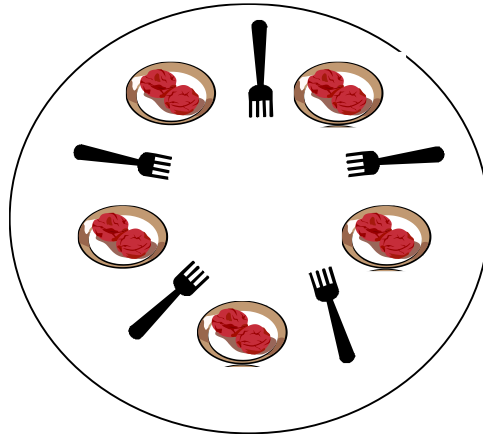


Bônus (Unidade III)

- 1) Fazer um programa, usando *threads*, para ilustrar o problema clássico do **Jantar dos Filósofos** (ver livro texto da disciplina). O desenho abaixo ilustra este cenário.



Em frente de cada prato existe um filósofo sentado, ou seja, existem 5 filósofos sentados à mesa. Para comer o seu espaguete, cada filósofo precisa de dois garfos. O comportamento do filósofo se resume a **comer** e **pensar**. Desta forma, aleatoriamente, o filósofo deve comer e pensar, indefinidamente. A ação de **comer** significa pegar dois (2) garfos, primeiro o do lado esquerdo e depois do lado direito, um em cada mão, e aguardar um determinado intervalo de tempo aleatório (`rand(3)`). Posteriormente, o filósofo deve deixar os garfos sobre a mesa, novamente primeiro o do lado esquerdo e depois do lado direito, a fim de pensar um pouco. A ação de **pensar** também deve ser implementada como uma inatividade por um determinado período de tempo aleatório.

Nesta implementação, propositalmente, você não deve usar qualquer mecanismo de exclusão mútua com o objetivo de experimentar o problema de inconsistência no uso dos recursos compartilhados.

- 2) Alterar o programa da questão 01 para incluir o uso de exclusão mútua entre *threads* (filósofos) vizinhas que compartilham o mesmo garfo. Utilizar o mecanismo de semáforos.
- 3) Como pode ser observado no resultado do programa #2, a implementação com semáforo resolverá o problema de compartilhamento de recursos, mas causará o problema de *deadlock*. Neste exercício você deverá fazer mudanças no seu código a fim de eliminar as condições de *deadlock*.