

# INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Licenciatura em Engenharia Eletrónica e  
Telecomunicações e de Computadores (LEETC)



**Sistemas Operativos**

**3º Trabalho**

**LT41D - Grupo 1**

Diogo Freixo Filipe (nº50387)

João Cardoso Santos (nº51009)

Rodrigo Roussado Viegas (nº50977)

## GRUPO I

1. O problema dos filósofos procura num cenário idealizado de forma a expor a dificuldade de evitar *deadlock* em situações de paralelismo. Consiste em imaginar vários filósofos sentados a uma mesa, cada um executa as seguintes ações (repetidamente):

1. Pegar no garfo à sua esquerda quando este estiver disponível.
2. Pegar no garfo à sua direita quando este estiver disponível.
3. Come por um período limitado (ocupando os dois garfos).
4. Pousa o garfo à sua direita na mesa.
5. Pousa o garfo à sua esquerda na mesa.

Numa primeira abordagem, pode parecer não haver qualquer problema, mas, pode dar-se o caso de todos os filósofos pegarem no garfo à sua esquerda, fazendo com que nenhum deles consiga pegar no garfo à sua direita, levando inevitavelmente a *deadlock*.

Alguns conceitos importantes que são trazidos à tona por este problema são:

**Deadlock** – Bloqueio do programa quando duas *threads* ficam em espera uma pela outra (por analogia, quando todos os filósofos ficam à espera que o que está à sua direita liberte o correspondente garfo esquerdo).

**Starvation** – Acontece quando uma *thread* é executada um número muito inferior de vezes quando comparada a outras (por analogia, é como um filósofo que nunca consegue agarrar nos garfos a tempo e, por consequência, não come).

Existe, ainda, a questão de tentar maximizar o paralelismo entre filósofos, evitando que um ou mais destes comam durante muito mais tempo que outros. Para isto, vamos avaliar algumas soluções para o problema:

**Solução Chandy/Mistra Simplificada:** esta solução consiste em fazer com que um ou mais filósofos (metade, por exemplo) peguem nos garfos pela ordem inversa dos outros. Apesar de não maximizar o paralelismo e poder causar *starvation*, é uma solução simples para o problema e, por isso, foi a escolhida por nós para implementar.

**Solução Hierárquica:** esta solução hierarquiza os garfos (através de um valor inteiro, por exemplo) e, cada filósofo, começa por pegar no garfo de menor valor. Esta solução não garante a inexistência de *starvation*.

**Solução por *Timeouts*:** esta solução faz com que, se um filósofo não conseguir obter acesso a ambos os garfos, larga aquele ao qual obteve acesso e espera um certo tempo antes de tentar ter acesso novamente. Esta solução também não garante que não haja *starvation* e depende fortemente do acaso de, quando acabar o *timeout*, haver garfos disponíveis.

Desta forma, podemos verificar que resolver problemas deste tipo, garantindo máximo paralelismo e evitando *starvation* é extremamente complicado.

## GRUPO II

Questões de Escolha Múltipla:

1. a) F  
b) V  
c) F  
d) V

2. a) F  
b) V  
c) V  
d) F