



## Trinco lógico

Sistemas Operativos – DEI - IST

1



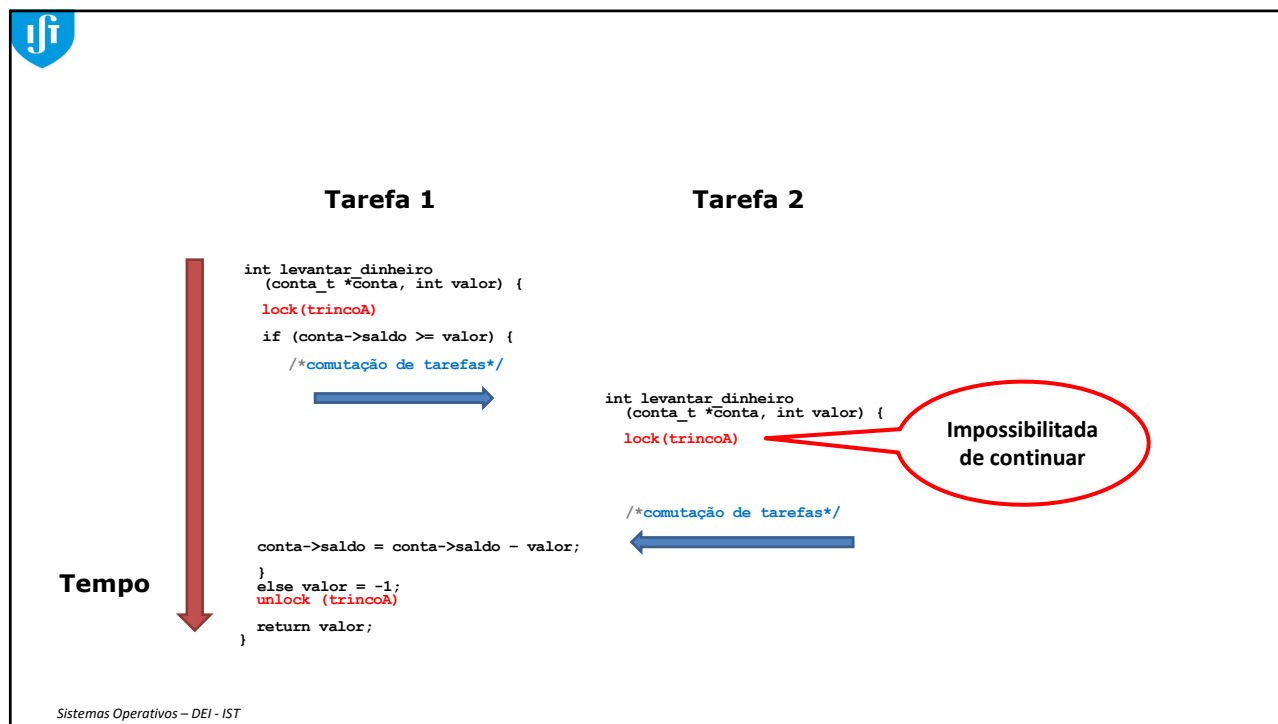
## Secção Crítica

```
int levantar_dinheiro (ref *conta, int valor)
{
    lock(trinco);
    if (conta->saldo >= valor) {
        conta->saldo = conta->saldo - valor;
    } else valor = -1;
    return valor;
}
```

} Secção crítica

Sistemas Operativos – DEI - IST

4



5

## Exclusão Mútua

- O exemplo anterior ilustra a propriedade de exclusão mútua

**Quando uma tarefa entra numa secção crítica mais nenhuma tarefa poderá entrar até o trinco ser libertado**

*Sistemas Operativos – DEI - IST*

6



## Propriedades desejáveis num trinco

- Propriedade de correção (*safety*)
  - Exclusão mútua - no máximo uma tarefa detém o trinco

Sistemas Operativos – DEI - IST

7



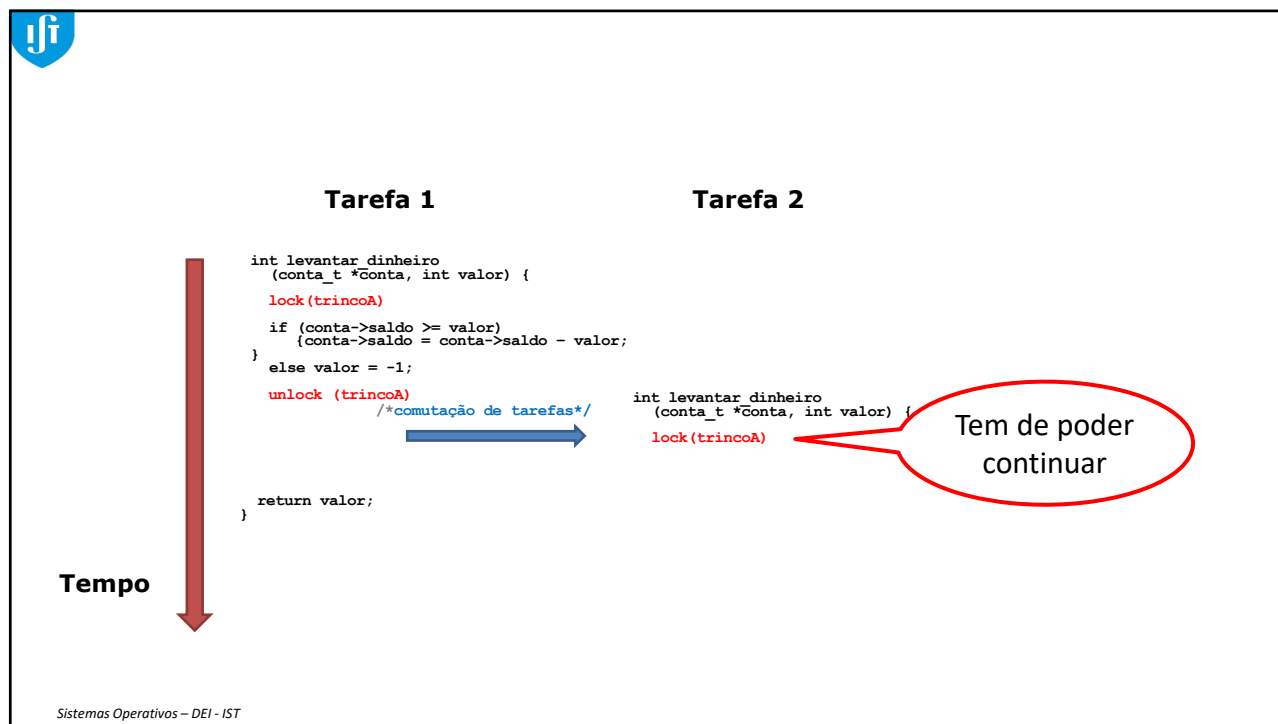
## Propriedade de progresso: ausência de interblocagem

Se nenhuma tarefa está a executar a secção crítica e existem tarefas que a pretendem executar, então uma delas tem de conseguir entrar num intervalo de tempo finito.

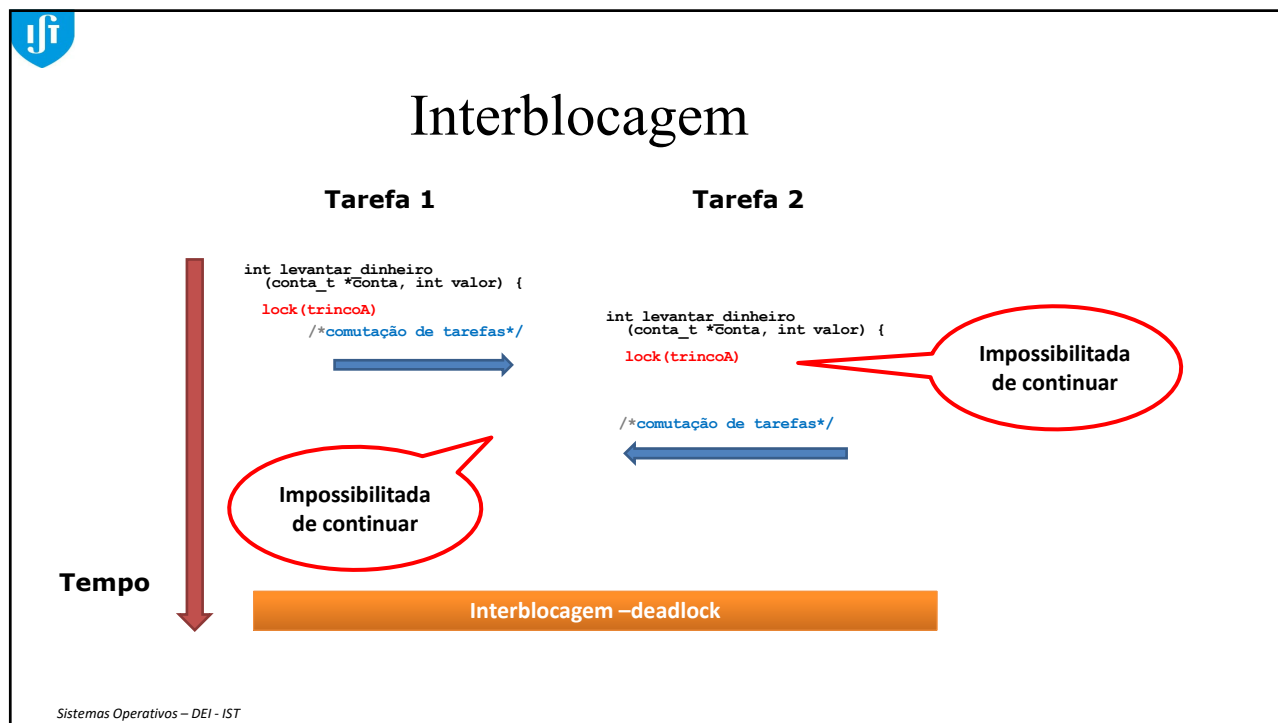
Esta condição visa impedir que uma secção crítica possa bloquear indefinidamente as tarefas entre si colocando-as numa situação de interblocagem (*deadlock*).

Sistemas Operativos – DEI - IST

8



9



10



## Propriedades desejáveis num trinco

- Propriedade de correção (*safety*)
  - Exclusão mútua
    - no máximo uma tarefa detém o trinco
- Propriedades de progresso (*liveness*)
  - Ausência de interblocagem (*deadlock*)
    - Se pelo menos uma tarefa tenta obter o trinco, então alguma o obterá (dentro de um tempo finito)

Sistemas Operativos – DEI - IST

11



## Propriedades desejáveis num trinco

- Propriedade de correção (*safety*)
  - Exclusão mútua
    - no máximo uma tarefa detém o trinco
- Propriedades de progresso (*liveness*)
  - Ausência de interblocagem (*deadlock*)
    - Se pelo menos uma tarefa tenta obter o trinco, então alguma o obterá (dentro de um tempo finito)
  - Ausência de mingua (*starvation*)
    - Se uma dada tarefa tenta obter o trinco, essa tarefa conseguirá obtê-lo (dentro de um tempo finito)
  - Eficiência

Sistemas Operativos – DEI - IST

12



## Trinco Lógico

- Vimos as propriedades que se pretendem de um trinco lógico
- Agora temos de ter ideias de como se implementa este conceito

Sistemas Operativos – DEI - IST

13



## Implementações simplistas

- Cortar pela raiz a razão da comutação das tarefas impedindo as interrupções
- A comutação só é possível, se houver interrupções de outra forma o Despacho não é invocado
- *Lock* do trinco executa um *disable interrupt*

Sistemas Operativos – DEI - IST

14



## Inibir as Interrupções

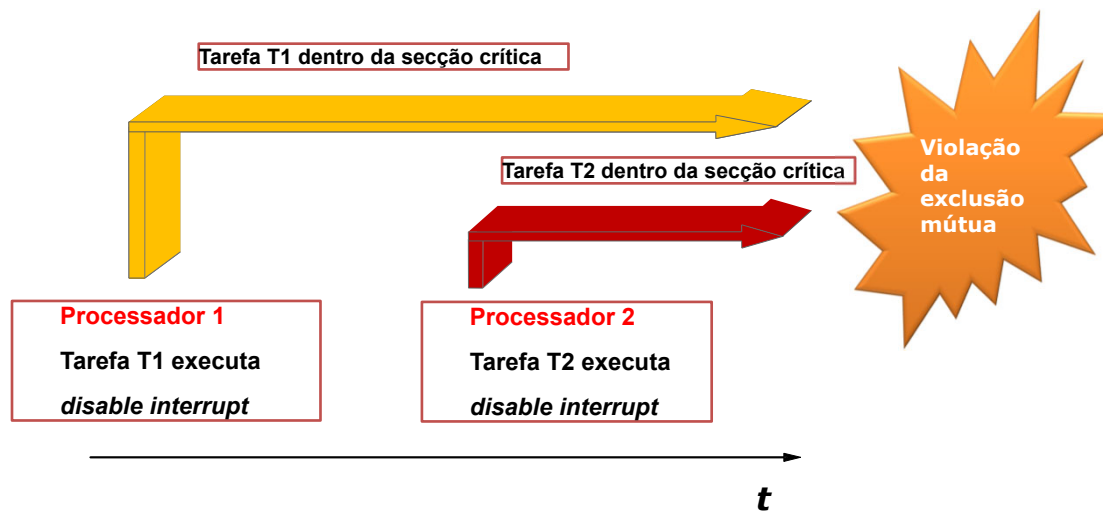
- As instruções de gestão das interrupções só podem ser executadas em **modo núcleo** do processador
- O sistema operativo ficaria vulnerável a secções críticas erradas que não libertassem a secção crítica
- **Não funciona em multiprocessadores**

Sistemas Operativos – DEI - IST

15



## A inibição das interrupções não funciona em multiprocessadores



Sistemas Operativos – DEI - IST

16



## Conclusões

- O trico permite garantir que uma secção de código se executa em exclusão mútua
- As secções críticas tem um conjunto de propriedades formais que permitem avaliar se uma dada implementação do trinco está correta
- Soluções simplistas, como bloquear as interrupções não seriam eficazes e estão erradas num ambiente de multiprocessamento