

# Sistemas Hardware-Software

Aula 21 - Semáforos II e Modelos de Concorrência

**Engenharia**

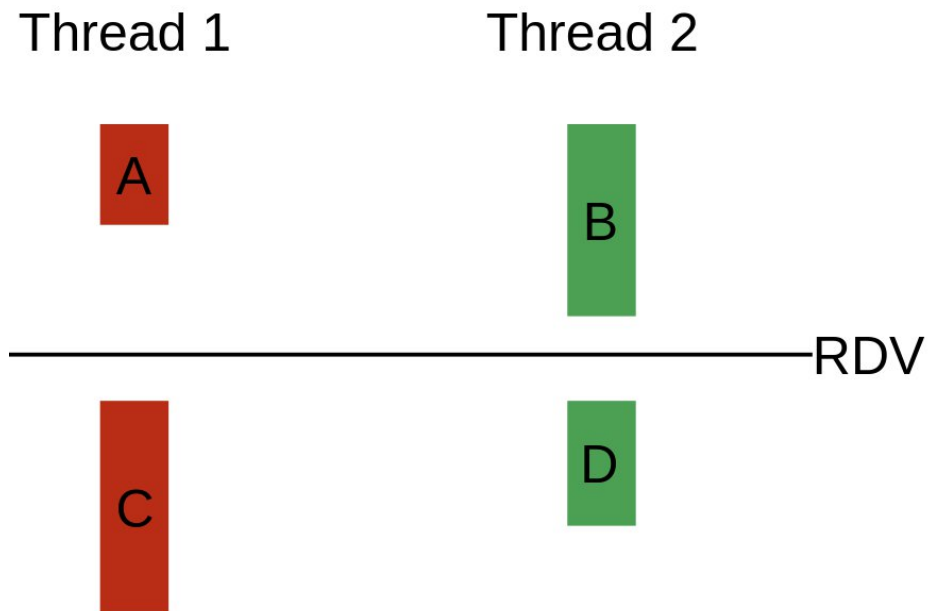
Fabio Lubacheski

Maciel C. Vidal

Igor Montagner

Fábio Ayres

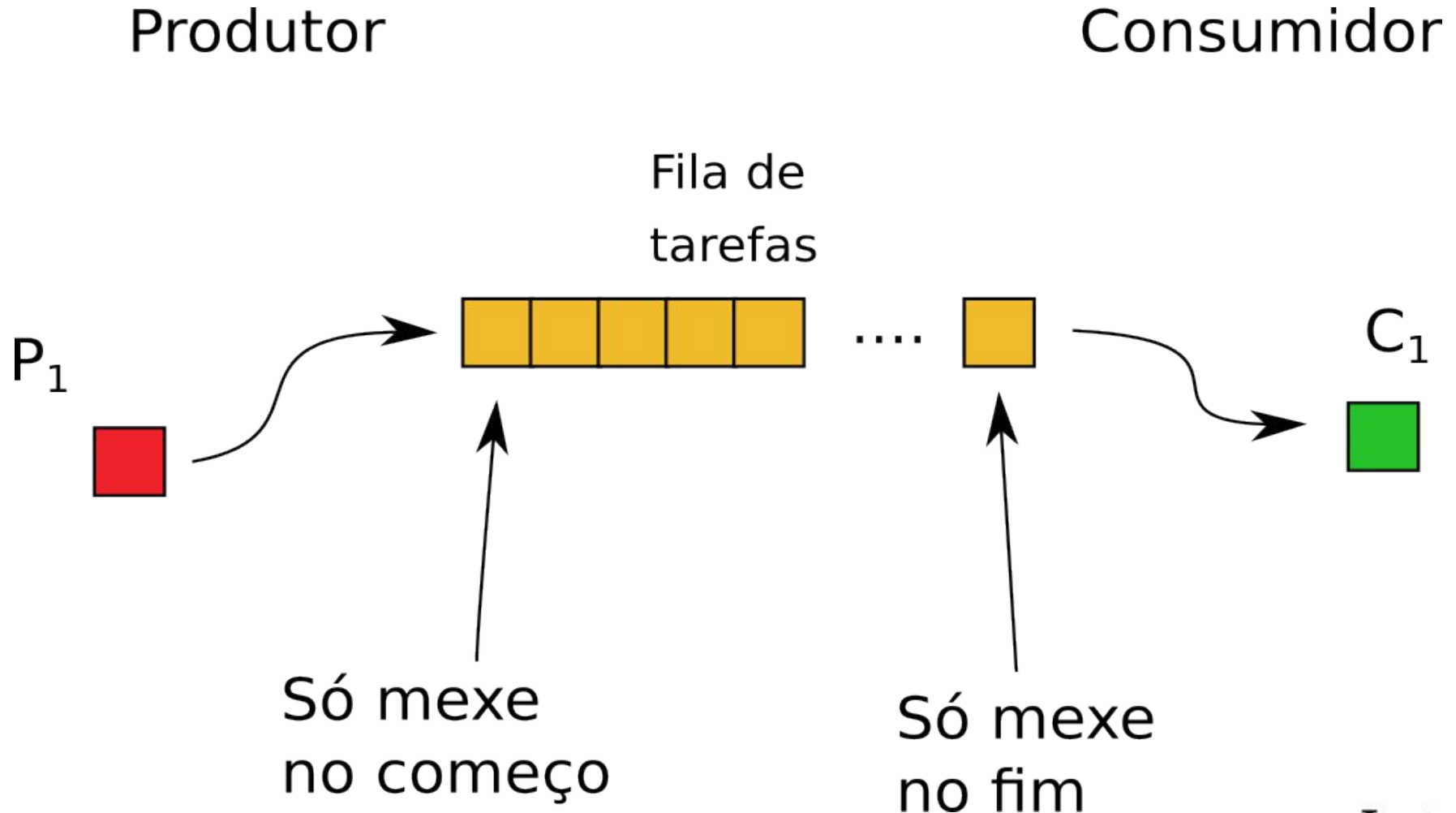
# Relembrando RDV



# Correção

**Veja a correção do problema de RDV da aula passada usando semáforos**

# Problema do Produtor/Consumidor



# Semáforos contadores

"Inteiro especial que nunca fica negativo"

Só pode ser manipulado por duas operações atômicas

## **Post:**

- Aumenta o valor em 1

## **Wait:**

- Se for positivo, diminui em 1
- Se for 0 fica esperando;

# Semáforos – Representação Conceitual

```
Wait(semaforo s)
{
    if( s > 0 )
        s = s - 1
    else
        "Bloqueia a thread"
}
```

```
Post(semaforo s)
{
    if("existe uma thread
        bloqueada" )
        "desbloqueia a
        thread"
    else
        s = s + 1
}
```

# Produtor/Consumidor (**pseudocódigo**)

```
// semaforos compartilhados
```

```
Mutex      mutex = 1;
```

```
Semaforo cheio = 0;
```

```
Semaforo vazio = N; // tamanho do buffer
```

```
produtor() {  
    while (true) {  
        item = produzir();  
        Wait(vazio);  
        Lock(mutex);  
        inserir(item);  
        Unlock(mutex);  
        Post(cheio);  
    }  
}
```

```
consumidor() {  
    while (true) {  
        Wait(cheio);  
        Lock(mutex);  
        item = retirar();  
        Unlock(mutex);  
        Post(vazio);  
        consumir(item);  
    }  
}
```

# Correção

**Veja a implementação do problema do Produtor/Consumidor.**

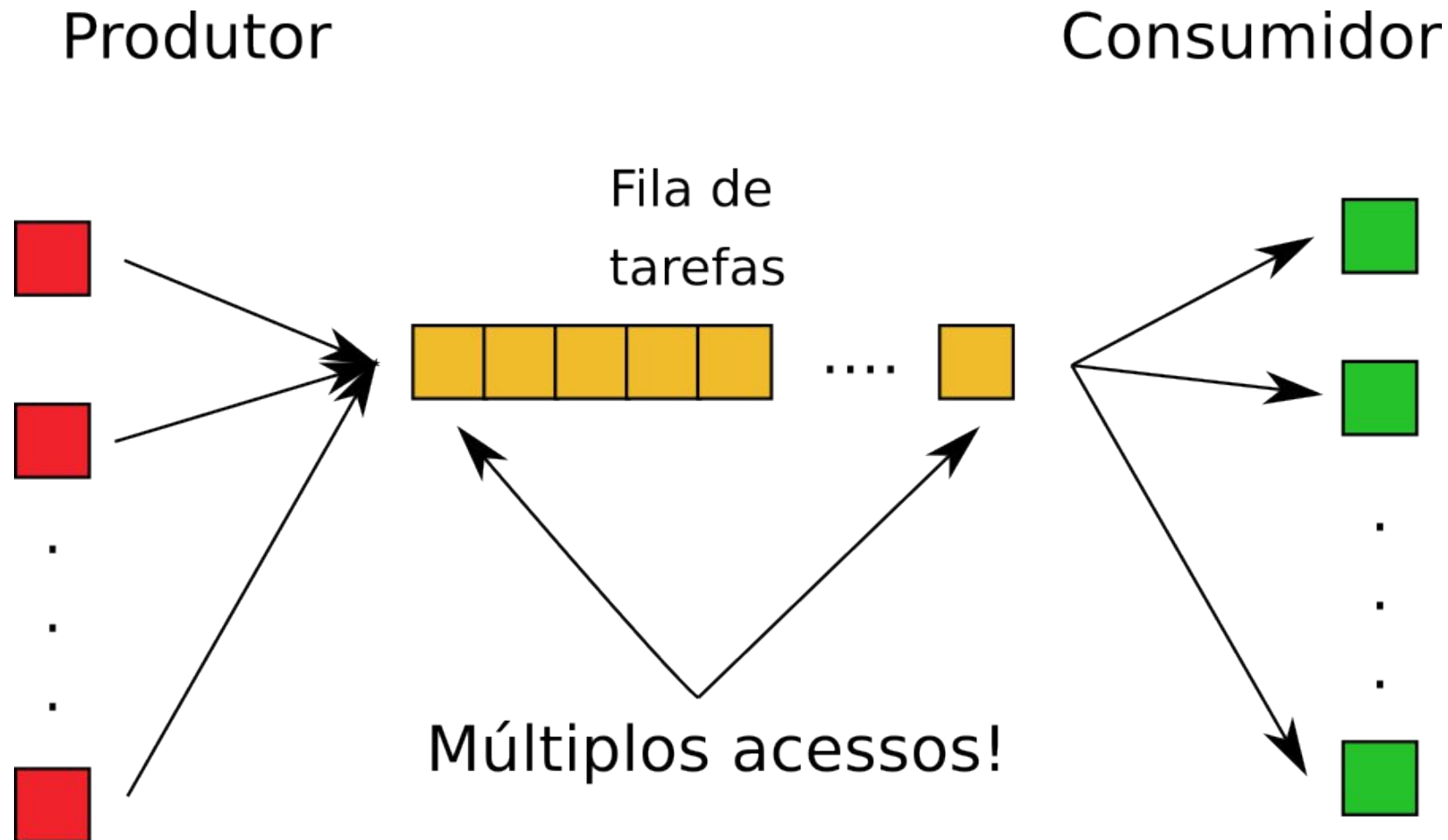


# Atividade prática

## Aplicação de Semáforos POSIX (20 minutos)

1. Limitar a **N** threads o acesso a um recurso

# Problema do **N** Produtores/Consumidores



# Insper

[www.insper.edu.br](http://www.insper.edu.br)