Insper

SuperComputação

Aula 15 – Produtor/Consumidor II, Leitores/Escritores

2018 – Engenharia

Igor Montagner, Luciano Soares <igorsm1@insper.edu.br>

Aulas passadas

1) Introdução a sincronização

2) Threads rodam programas diferentes

Exclusão mútua - Mutex

- Acessamos um recurso compartilhado
- Região crítica: porção do código que manipula o recurso
- Propriedade: somente um thread por vez na região crítica

Variável de Condição

- Preciso de
  - Acesso Exclusivo (Mutex)
  - Condição seja verdadeira
- Útil para sincronizar progresso de threads de maneira condicional

Variável de Condição - Wait

wait (m, P): espera ter o mutex e a condição P ser verdadeira

```
while !P
    release(m)
    wait(m)
```

- Se a condição for verdadeira na primeira checagem prossegue
- Se não espera ser notificado para checar de novo

Variável de Condição - Notify

- notify_one: notifica uma thread que a condição P se tornou verdadeira

- notify_all: notifica todas as threads que a condição P se tornou verdadeira

Duas filas:

1) Threads esperando serem notificadas para checarem a condição

2)Threads esperando o mutex

Semáforo

Inteiro especial com duas funções

- Up() → soma um no valor
- Down() → se valor > 0: valor -= 1
  senão: espera até que valor > 0

Se inicializado com S, permite até S acessos concorrentes

Semáforo

- Generalização do mutex
- Pode ser construído usando Mutex e Variável de Condição
- Representa o número de unidades de um recurso
    - supondo que Up() seja seguido de Down()

Hoje

1) Dúvidas da atividade 2

2) Modelo produtor consumidor M-N

3) Modelo Leitores e Escritores

Hoje

1) Dúvidas da atividade 2 – partes 1 e 2

2) Modelo produtor consumidor M-N

3) Modelo Leitores e Escritores

Modelo produtor-consumidor

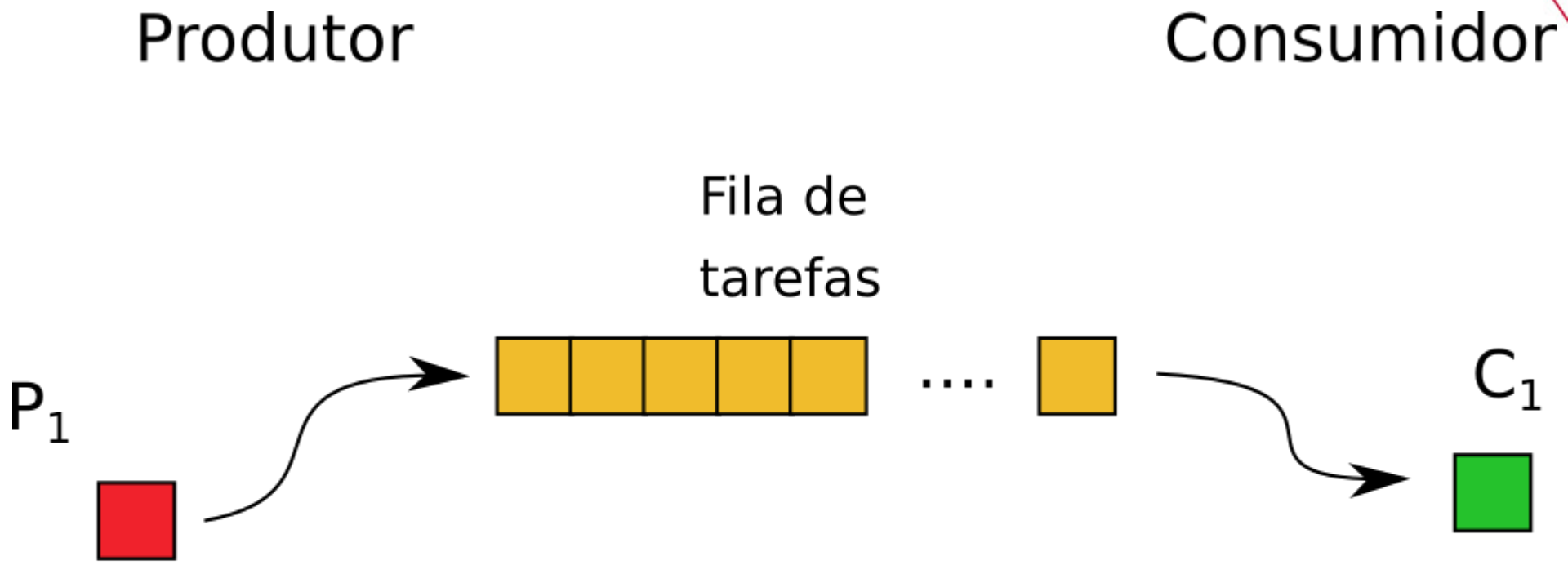
Dois conjuntos de threads

- Produzem tarefas a serem executadas
  - pode depender de um recurso compartilhado
  - controlar tamanho das tarefas
- Consomem as tarefas e as executam
  - tarefas independentes entre si
  - tarefas independentes da produção

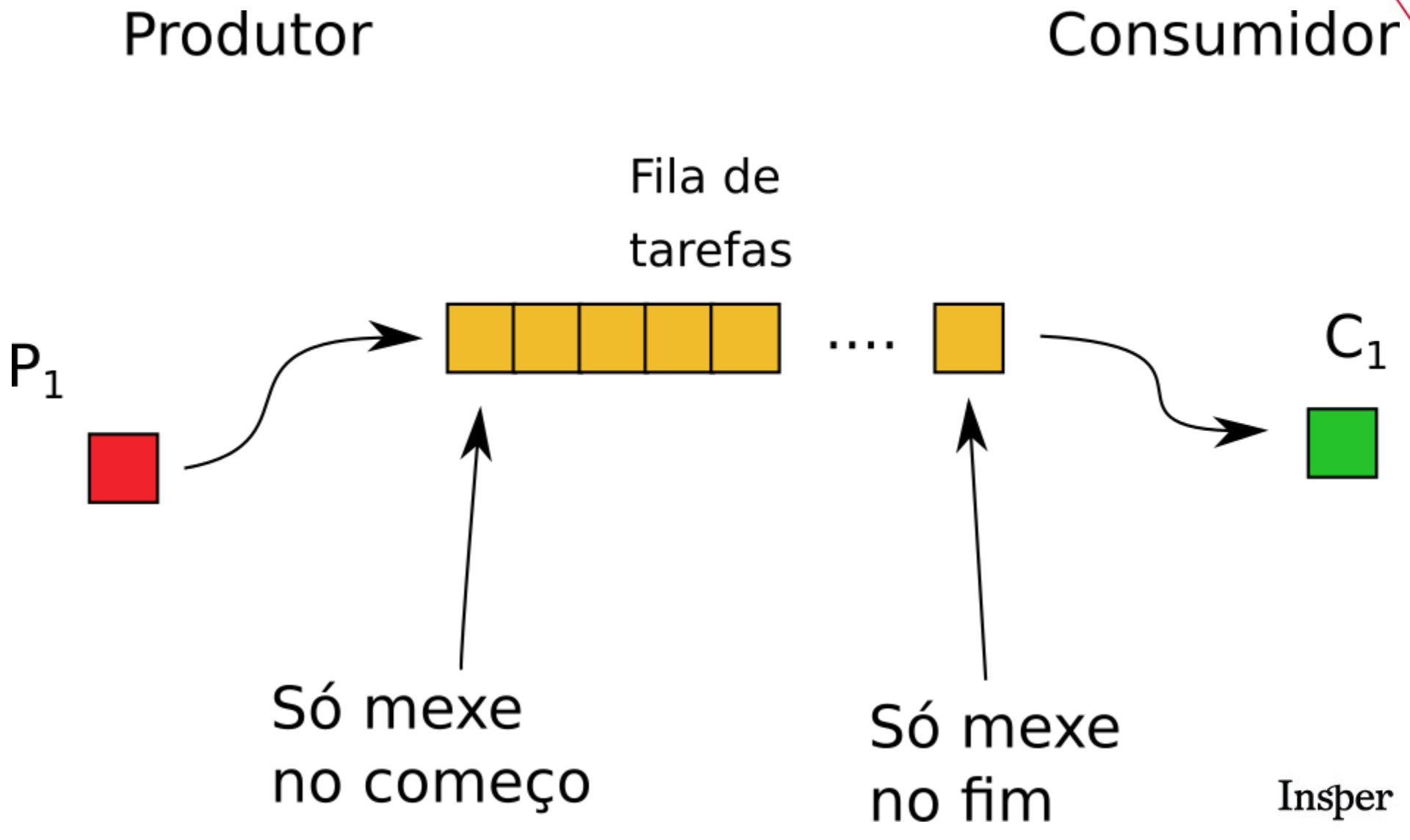
Modelo produtor-consumidor

- Depende de uma fila compartilhada
- Consumidor retira tarefas da fila
- Produtor adiciona tarefas à fila

Modelo produtor-consumidor 1-1



Modelo produtor-consumidor 1-1

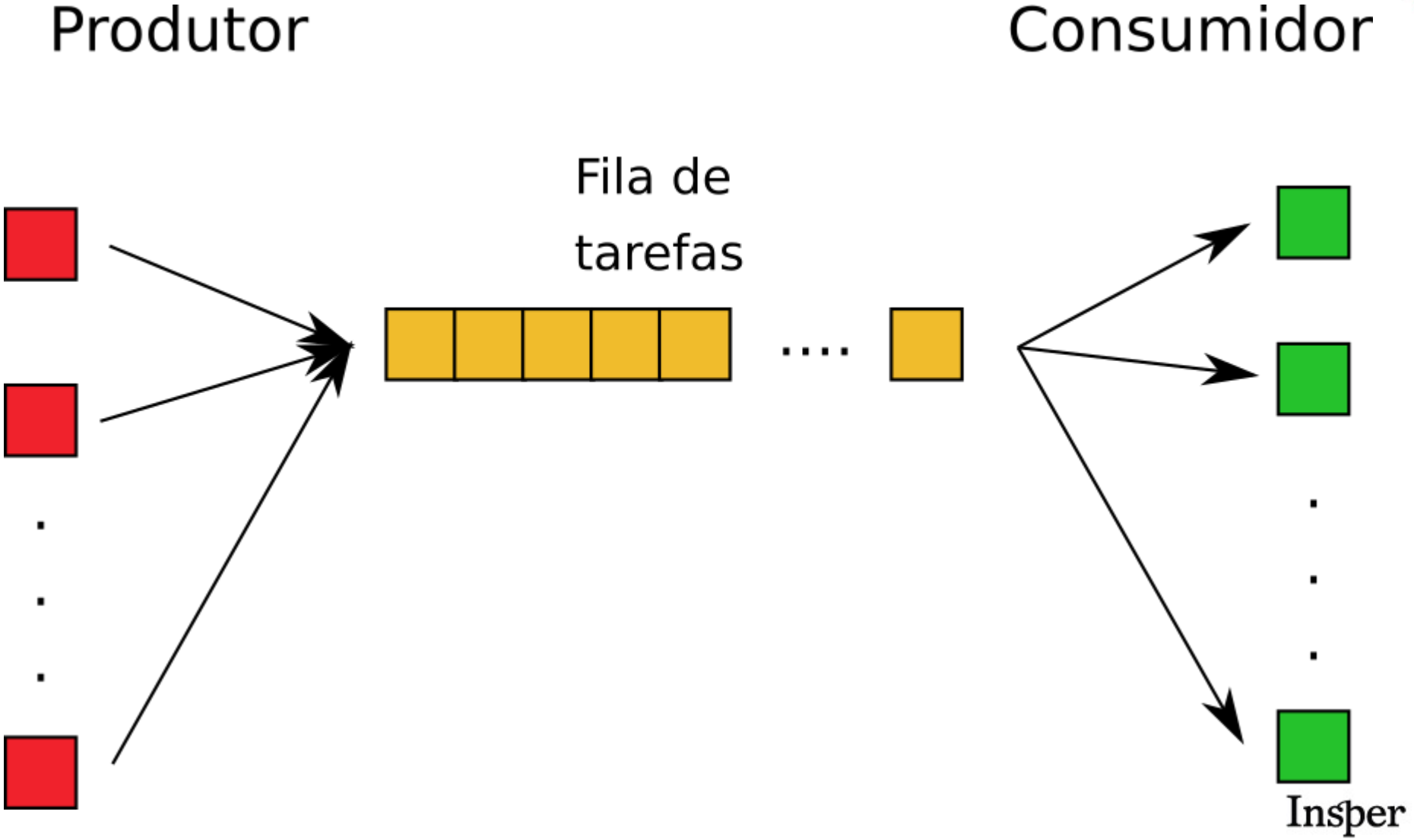


Modelo produtor-consumidor 1-1

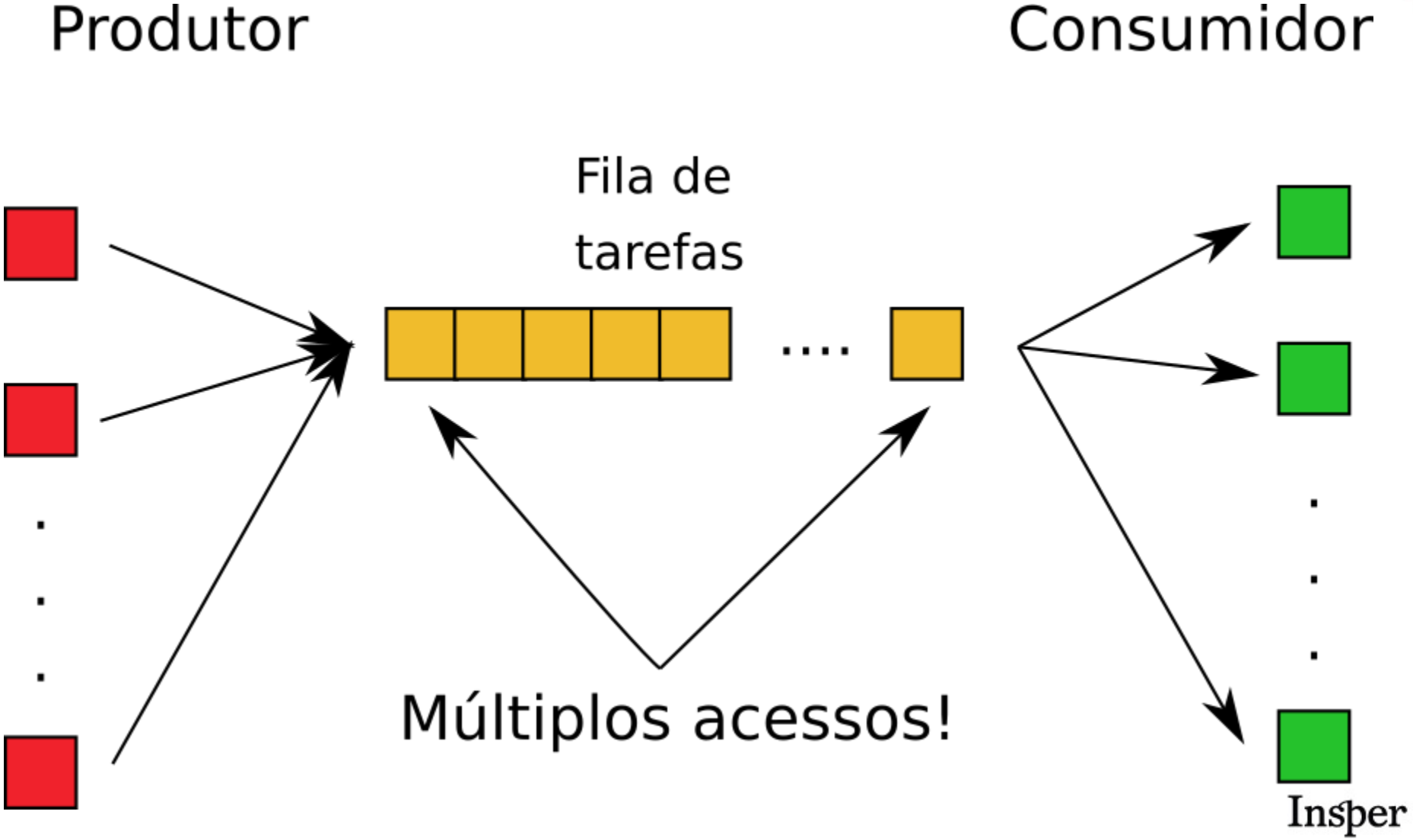
Dada a implementação de FilaInfinita,

- É necessário sincronizar o acesso? Quando?
- Quais operações são paralelas?

Modelo produtor-consumidor M-N



Modelo produtor-consumidor M-N



Modelo produtor-consumidor M-N

Dada a implementação de FilaInfinita,

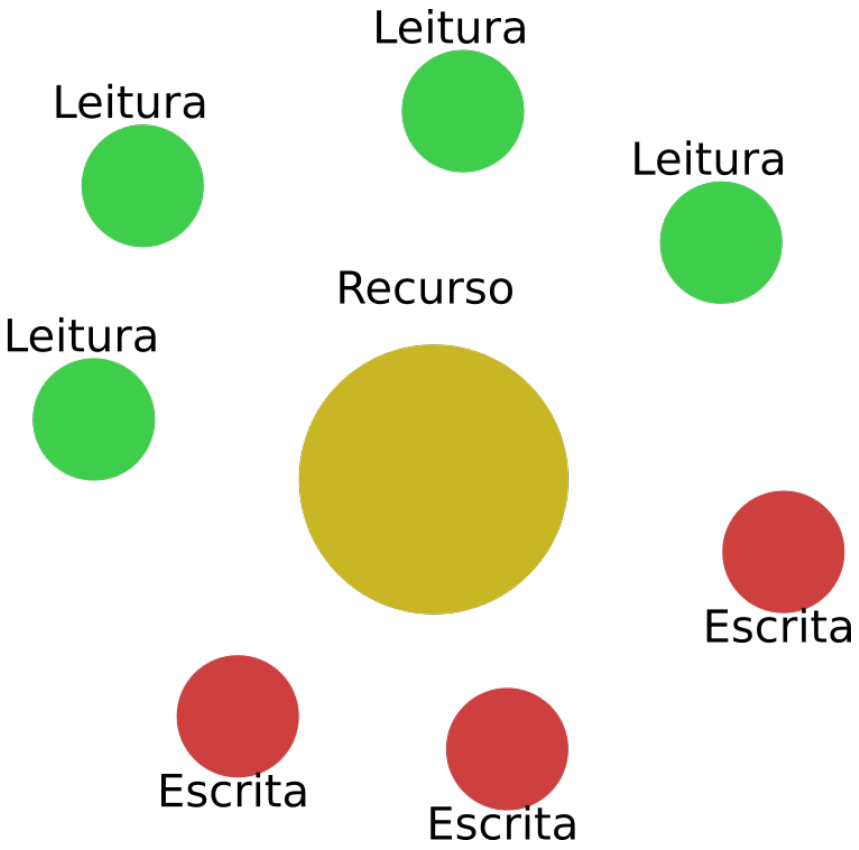
- É necessário sincronizar o acesso? Quando?
- Quais operações são paralelas?

Modelo produtor-consumidor M-N

Dada a implementação de **FilaFinita**,

- É necessário sincronizar o acesso? Quando?
- Quais operações são paralelas?
- A implementação da fila possui dois *assert*. Eles são necessários?

Modelo Leitores e Escritores



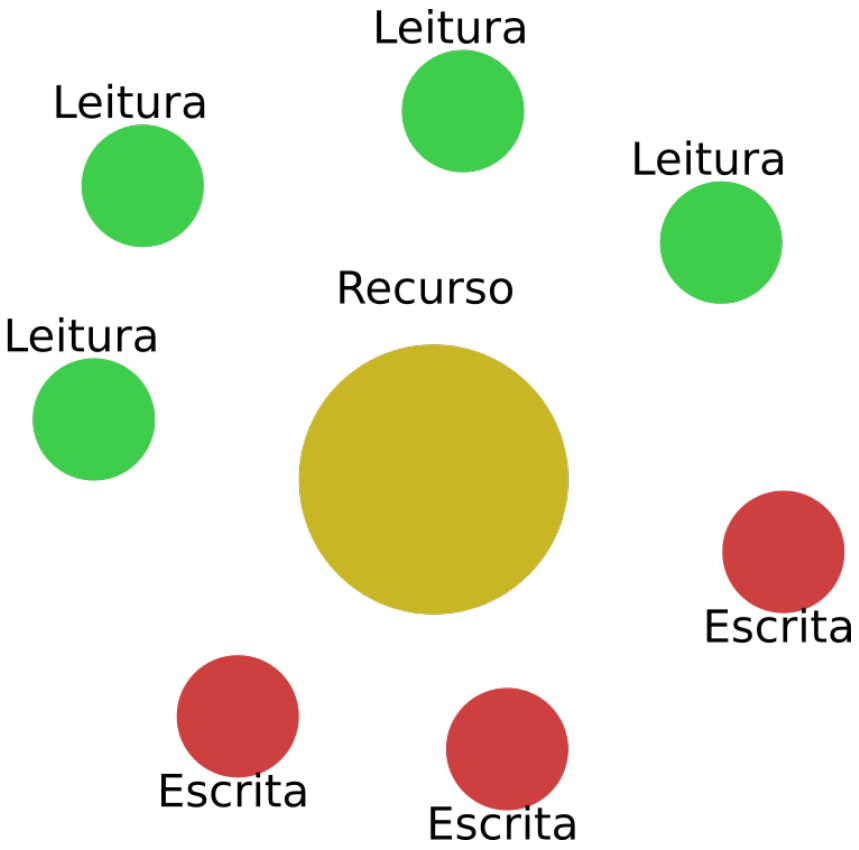
Um recurso compartilhado por vários processos que

- **Leem** o estado do recurso
- **Modificam** o estado do recurso

Com as restrições

- **Leituras** podem ser feitas simultâneamente
- **Escritas** necessitam de acesso exclusivo

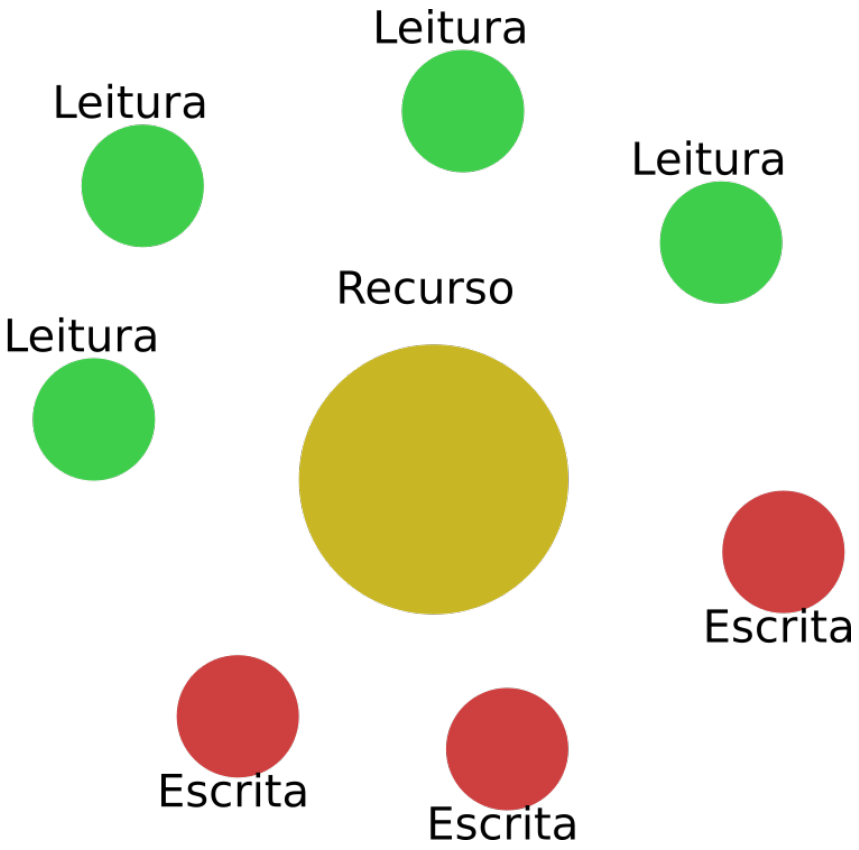
Modelo Leitores e Escritores



Problemas:

- O quê acontece se a frequência de leitores é alta e a frequência de escritores é baixa?
- E se for o oposto?

Modelo Leitores e Escritores

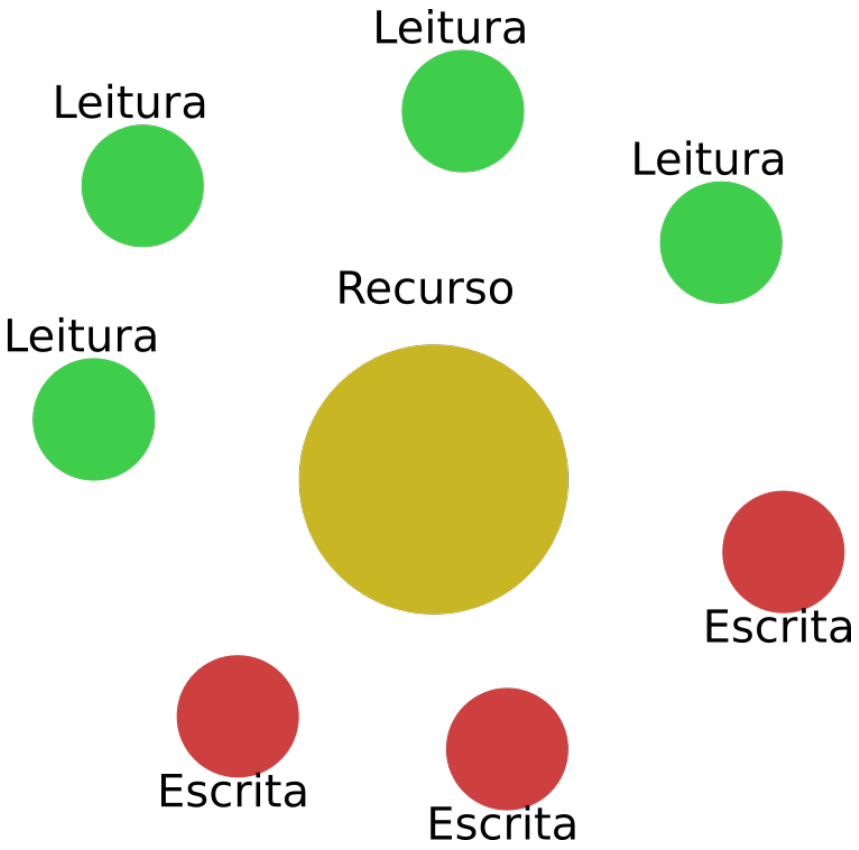


Problemas:

- O quê acontece se a frequência de leitores é alta e a frequência de escritores é baixa?
- E se for o oposto?

Starvation: situação onde uma thread (ou grupo de threads) nunca consegue acesso a um recurso.

Modelo Leitores e Escritores



Atividade: três grupos irão propor, cada um, uma solução onde

1) Leitores tem preferência, não há ordem garantida;

2) Escritores tem preferência, não há ordem garantida;

3) Os acessos são feitos por ordem de chegada, mas se há vários leitores em seguida eles podem executar simultâneamente;

Referências

- Livros:
  - Hager, G. ; Wellein, G. **Introduction to High Performance Computing for Scientists and Engineers**. 1ª Ed. CRC Press, 2010.
- Artigos:
  - Duran, Alejandro, Julita Corbalán, and Eduard Ayguadé. "Evaluation of OpenMP task scheduling strategies." In *International Workshop on OpenMP*, pp. 100-110. Springer, Berlin, Heidelberg, 2008
- Internet:
  - https://www.youtube.com/playlist?list=PLLX-Q6B8xqZ8n8bwjGdzBJ25X2utwnoEG
  - http://www.openmp.org/wp-content/uploads/omp-hands-on-SC08.pdf
  - http://extremecomputingtraining.anl.gov/files/2016/08/Mattson_830aug3_HandsOnIntro.pdf

Insper

www.insper.edu.br