

# Programação Concorrente

**Condições de Corrida,  
Regiões Críticas e  
Exclusão Mútua**

Prof. Eduardo Alchieri

# Processos

## (comunicação entre processos)

- Os processos executando concorrentemente podem ser de dois tipos:
- Independentes
  - Um processo é independente se não puder afetar ou ser afetado pelos outros processos em execução no sistema
  - Qualquer processo que não compartilhe dados com qualquer outro processo é independente
- Cooperativos
  - Um processo é cooperativo se puder afetar ou ser afetado por outros processos em execução no sistema
  - Qualquer processo que compartilhe dados com outros processos

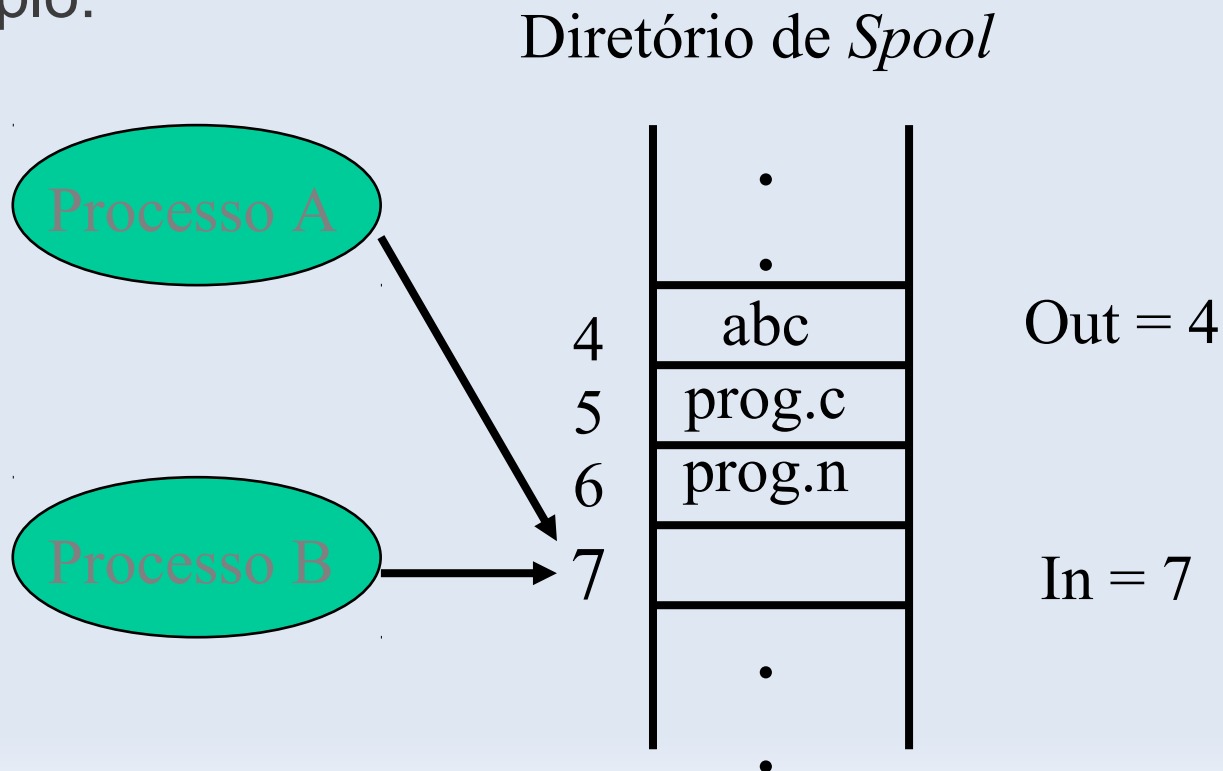
# Processos

## (comunicação entre processos)

- Os processos cooperativos precisam de mecanismos de comunicação entre processos (***Interprocess Communication – IPC***) que lhe permitam a troca de dados
- Existem dois modelos fundamentais de comunicação entre processos:
  - Memória compartilhada
    - É estabelecida uma área compartilhada para que os processos possam ler ou escrever nessa área
  - Memória distribuída
    - A comunicação é toda feita através de troca de mensagens

# Condições de Corrida

- Problema: Condições de Corrida (race conditions)
  - Condições de corrida são situações onde dois ou mais processos estão acessando dados compartilhados, e o resultado final do processamento depende de quem executa e quando executa
- Exemplo:

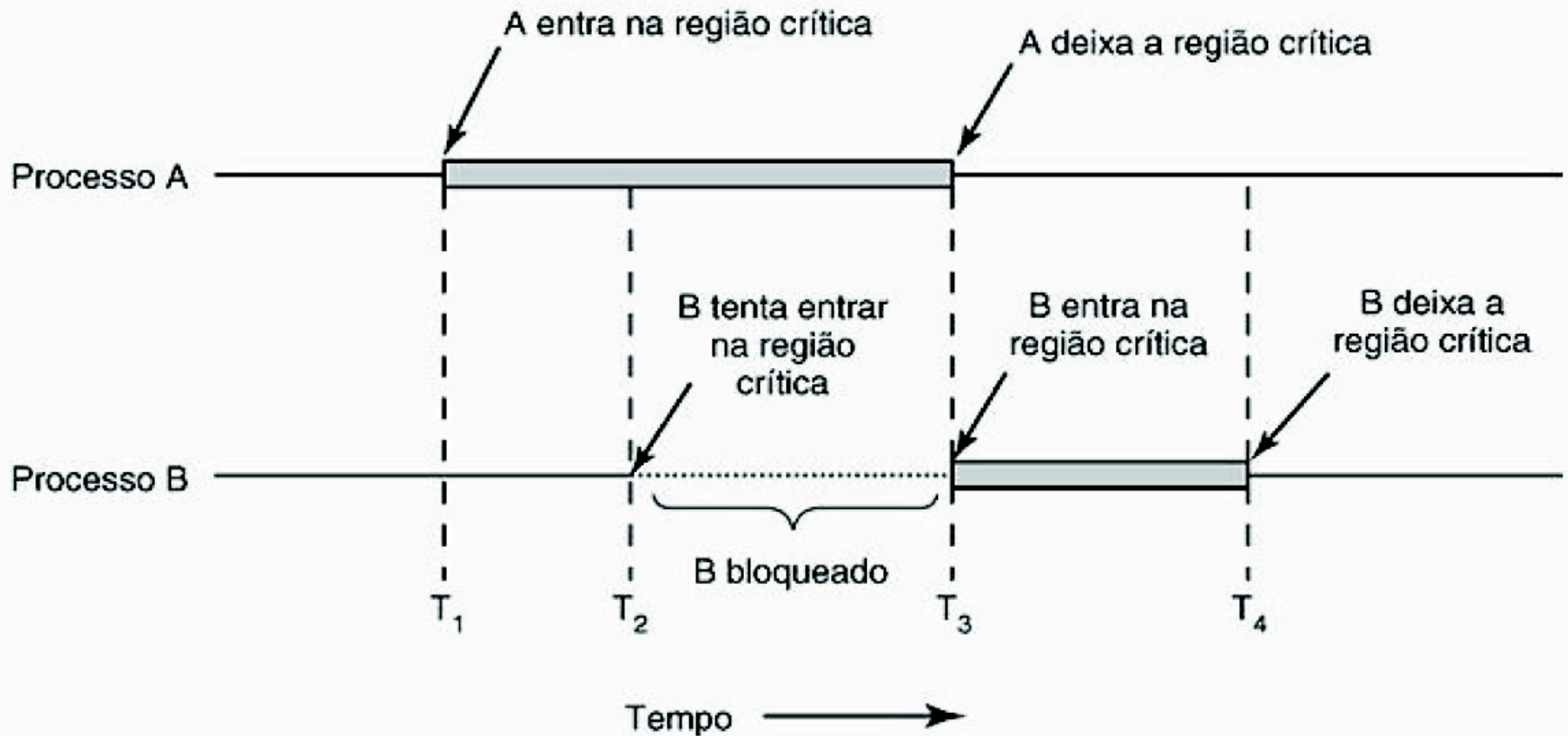


# Condições de Corrida

- Como evitar condições de disputa ?
  - Encontrar alguma forma de proibir que mais de um processo acesse o dado compartilhado ao mesmo tempo, isto é, estabelecer a **exclusão mútua** de execução
- **Exclusão mútua**: impedir que dois ou mais processos acessem um mesmo recurso ao mesmo tempo.
- **Região Crítica**: parte do código do programa onde é feito o acesso à memória compartilhada (ou ao recurso compartilhado), ou seja, é a parte do programa cujo processamento pode levar à ocorrência de condições de corrida

# Exclusão Mútua

- Exclusão mútua usando regiões críticas



# Região Crítica

- Fluxo de execução de um processo
  - Executa região não crítica
  - Executa "procedimento" para entrar na região crítica
  - Executa sua REGIÃO CRÍTICA
  - Executa "procedimento" para deixar a região crítica
  - Executa região não crítica

# Região Crítica

- Uma boa solução (correta e eficiente) para a condição de corrida requer que quatro condições sejam satisfeitas:
  - Dois ou mais processos não podem estar simultaneamente dentro de suas regiões críticas correspondentes
  - Nenhuma consideração pode ser feita a respeito da velocidade relativa dos processos, ou a respeito do número de processadores disponíveis no sistema
  - Nenhum processo que esteja executando fora de sua região crítica pode bloquear a execução de outro processo
  - Nenhum processo pode ser obrigado a esperar indefinidamente para entrar em sua região crítica



# Soluções para Exclusão Mútua

- **Desabilitar Interrupções**

- Consiste em desabilitar as interrupções de cada processo logo após o seu ingresso em uma região crítica, habilitando-as outra vez imediatamente antes de sair da região crítica

- **PERIGO!!!**

- Não é uma boa prática atribuir aos processos de usuários o poder de desabilitar interrupções, interferindo diretamente no núcleo do sistema operacional
- Além disso, em computadores com mais de uma CPU, desabilitar as interrupções de uma CPU não impede que outras CPUs interfiram nas operações (acesse a memória compartilhada) que a primeira CPU está executando