

Sistemas Operacionais

Comunicação Interprocessos
Programação Concorrente

Sumário

- 1 Definição
- 2 Comunicação
- 3 Sincronismo
- 4 Mensagens

- 5 Condição de Corrida
- 6 Desativar Interrupções
- 7 Variável Lock
- 8 Alternância
- 9 Bibliografia

Definição

- › Sistemas Distribuídos;
- › Programação Paralela;
- › Pipeline;
- › Modularidade;
- › Multiprocessamento;
- › Multiusuário;
- › Interatividade (Interface Gráfica);

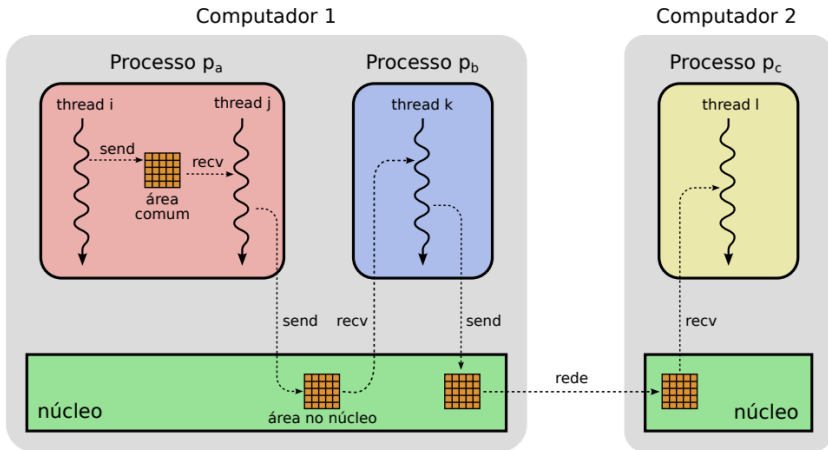


Figura 1: Intercomunicação de Processos (Maziero, 2019)

Questões em IPC:

- 1 Como um processo pode passar informações para outro.
- 2 A segunda tem a ver com certificar-se de que dois ou mais processos não se atrapalhem;
- 3 A terceira diz respeito ao sequenciamento adequado quando dependências estão presentes.

É importante mencionar que duas última dessas questões aplicam-se igualmente bem aos threads.

(TANENBAUM, 2010)

Comunicação

› Comunicação Direta:

- » Há destino definido.
- » Pouco utilizada.
- » Exemplo: comando `kill <PID>`

› Comunicação Indireta:

- » Há origem definida.
- » Utilização comum.
- » Exemplo: um processo "ouvindo" na porta 80.
- » Utiliza um canal (protocolos de rede).

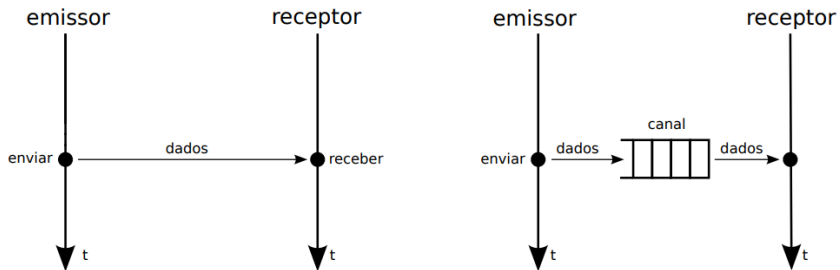


Figura 2: Comunicação direta e indireta (Maziero, 2019)

Sincronismo

- › Comunicação Síncrona:
 - » Bloqueante.
- › Comunicação Assíncrona:
 - » Não Bloqueante;
 - » Não é comum (A assincronia não está relacionada com o canal).
- › Comunicação Semi- Síncrona:
 - » De utilização comum.
 - » Operações bloqueantes.
 - » *Timeout*.
 - » Utilizada quando envolve envio de dados pela rede.

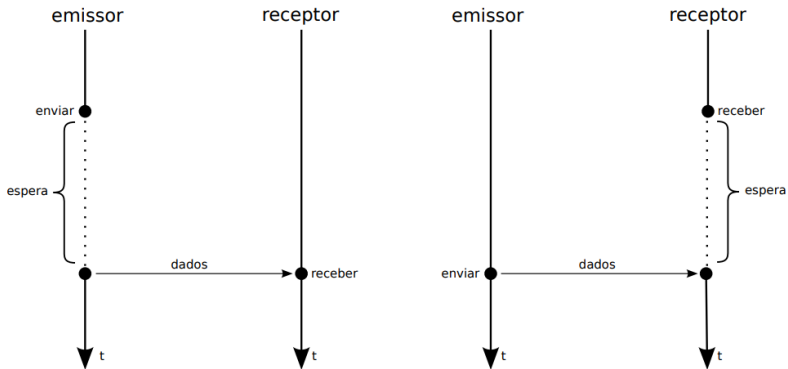


Figura 3: Comunicação bloqueante (Maziero, 2019)

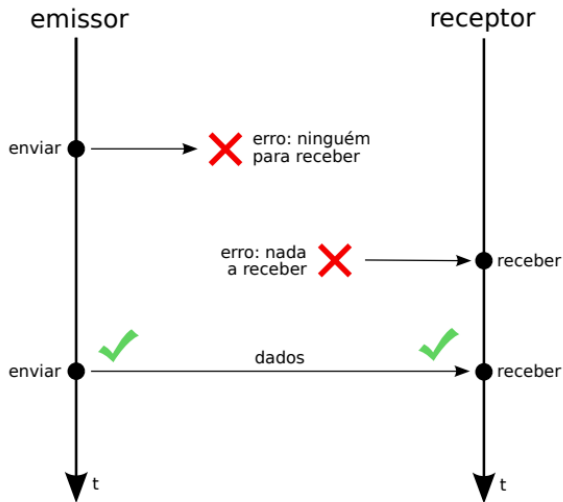


Figura 4: Comunicação não bloqueante (Maziero, 2019)

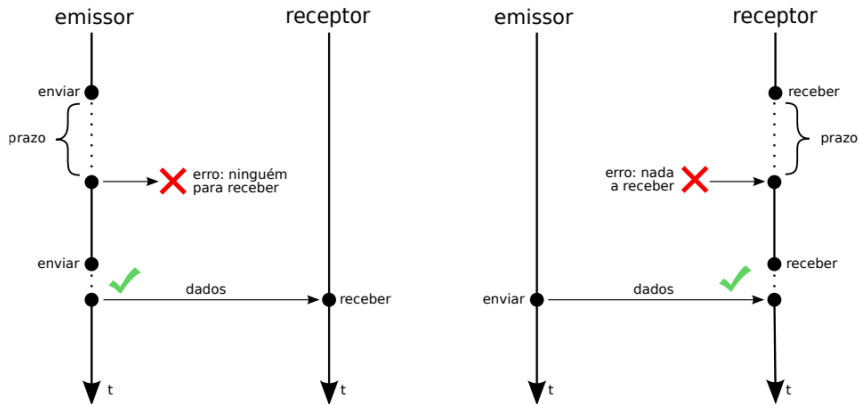


Figura 5: Comunicação semi-bloqueante (Maziero, 2019)

Mensagens

- Pacotes de dados:
 - Cada parte da mensagem é independente.
 - Protocolo TCP ou UDP.
- Fluxo de Dados:
 - Arquivos de bytes.
 - Pipes.

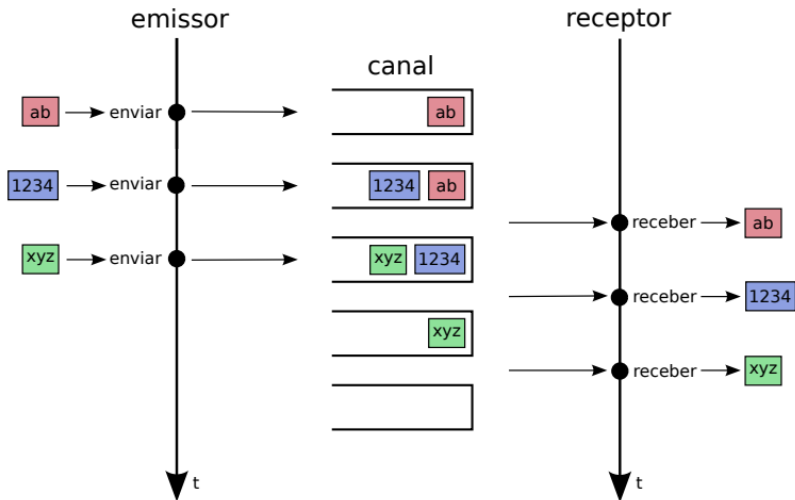


Figura 6: Mensagens (Maziero, 2019)

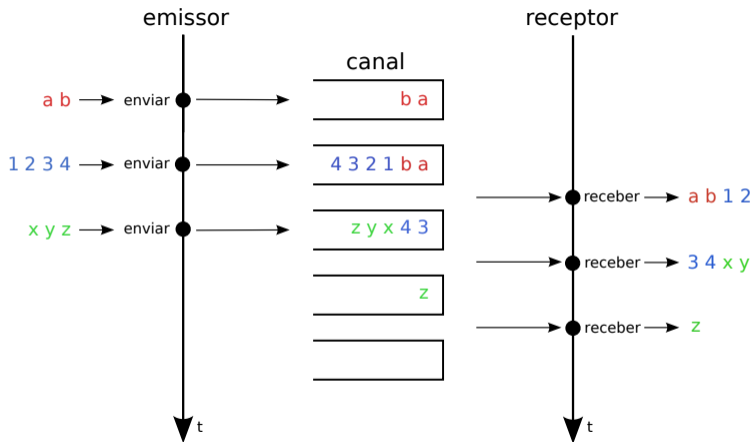


Figura 7: Fluxo de dados (Maziero, 2019)

Problema Produto Consumidor I

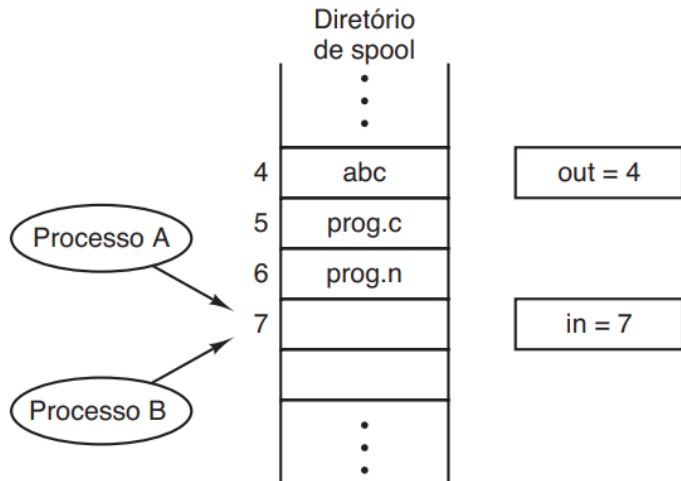
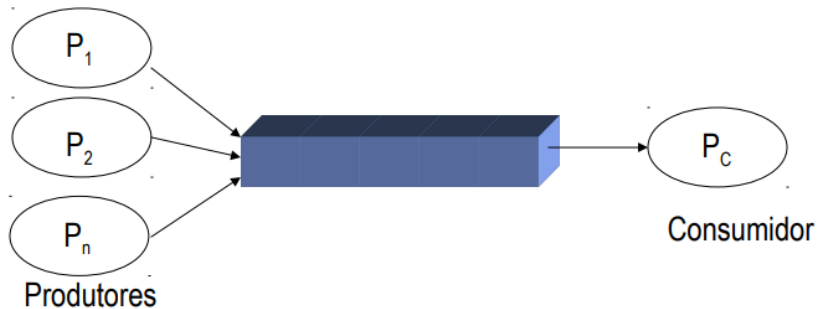


Figura 8: Processos acessando memória compartilhada (TANENBAUM, 2010)

Problema Produto Consumidor II



Outro Exemplo:

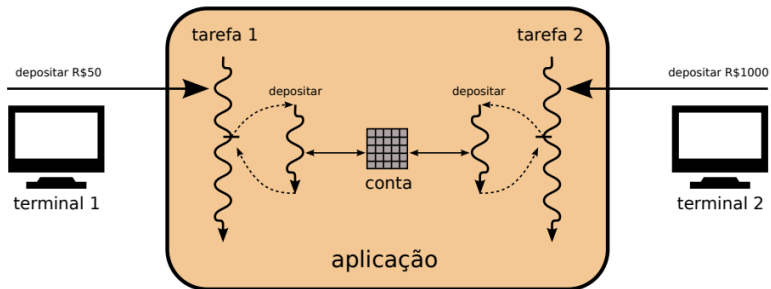


Figura 9: Exemplo: sistema bancário (Maziero, 2019).

Problema Produto Consumidor IV

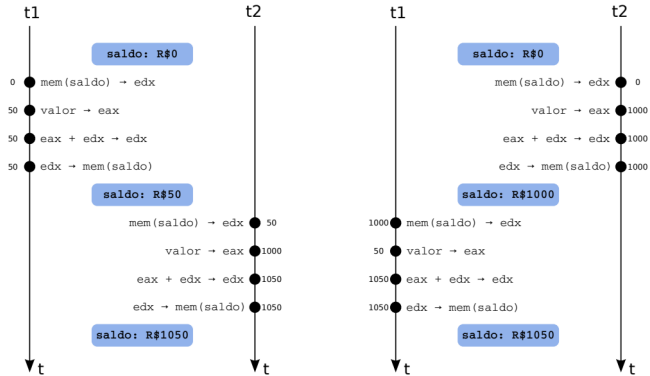
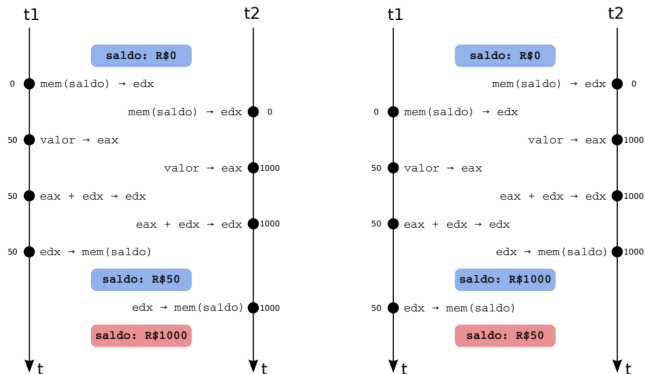


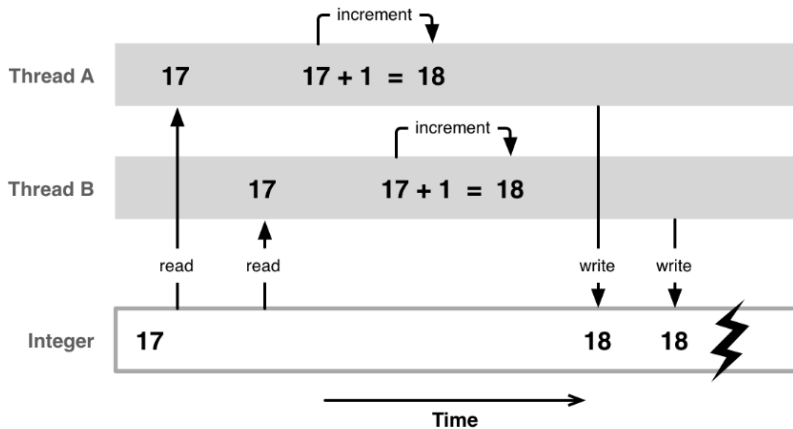
Figura 10: Exemplo: sistema bancário

Problema Produto Consumidor V



Exemplo de condição de corrida (Maziero, 2019)

Condição de Corrida



As **Condições de Bernstein** ajudam a **verificar se duas tarefas podem ser executadas em paralelo** sem causar **condições de disputa** (conflitos de acesso a variáveis).

- As tarefas são **independentes** se:
 - Não há **leitura** de valores ainda **não escritos**;
 - Não há **escrita** sobre a **mesma variável**;
- Portanto, **podem executar em paralelo com segurança**.

Seção Crítica

Trecho de código de cada tarefa que acessa dados compartilhados, onde podem ocorrer condições de disputa.

Exclusão Mútua

Impedir o entrelaçamento de seções críticas, de modo que apenas uma tarefa esteja na seção crítica a cada instante.

Propriedades da exclusão mútua

- 1 **Exclusão mútua:** Dois ou mais processos não podem estar simultaneamente em uma seção crítica.
- 2 **Espera limitada:** Nenhum processo deve esperar infinitamente para entrar em uma seção crítica.
- 3 **Independência de outros processos:** Nenhum processo fora da seção crítica pode bloquear a execução de um outro processo.
- 4 **Independência do hardware:** Não fazer considerações sobre o número de processadores, nem de suas velocidades relativas.

Desativar Interrupções

Interrupções

Inibir interrupções durante a sessão crítica.

Monoprocessado

- A CPU só é chaveada de um processo para outro em consequência de uma **interrupção de relógio** (preempção por tempo).

Multiprocessado

- Outras CPUs continuam executando, e seus processos podem acessar uma **região crítica**.
- Problema: e se uma CPU **desligasse as interrupções** e **nunca mais as ligasse de volta**?

Muito poder em nível de aplicação!

Variável Lock

Lock

Uma **variável de trava** (*lock*) é uma estrutura que **controla o acesso a recursos compartilhados**.

- › Garante que apenas **um processo ou thread** utilize o recurso por vez;
- › trancado (*locked*);
- › destrancado (*unlocked*);

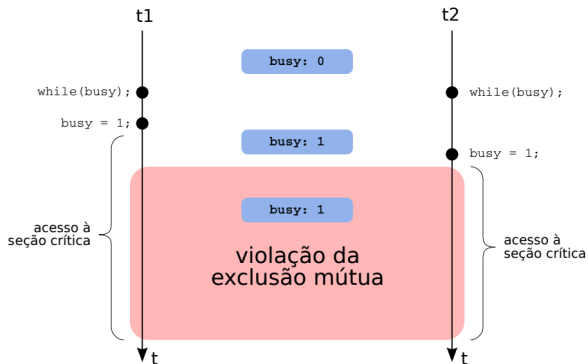


Figura 11: Problema da variável de trava (Maziero, 2019)

;

Alternância

Alternância - Espera Ocupada (*busy waiting*)

A variável **turno** indica de quem é a vez (*turn*) de entrar na **seção crítica**, garantindo que apenas **um processo por vez** possa acessar o recurso.

- **Sequência circular:** Rotação cíclica entre os processos, onde, ao fim do turno de um processo, o próximo processo na sequência recebe o direito de entrar na seção crítica.
- **Problema:** Se um processo, ao chegar sua vez, **não entrar na seção crítica**, pode ocorrer:
 - » **Inanição** (*starvation*);
 - » **Bloqueio desnecessário** de outros processos.

- › **Garante a ordem** de acesso à seção crítica;
- › **Garante a exclusão mútua**;
- › Porém, é **ineficiente**;
- › Os processos **ficam testando continuamente** se é o seu turno (*busy waiting*).

Combina a solução com variáveis *lock* e alternância.

Capítulo 3 (OLIVEIRA; CARISSIMI; TOSCANI, 2001)



Bibliografia



OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. **Sistemas Operacionais**. 2. ed. Porto Alegre: Sagra-Luzzatto, 2001.



STALLINGS, William. **Operating Systems: Internals and Design Principles**. 6. ed. Upper Saddle River, NJ: Prentice-Hall, 2009.



TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.



SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. **Sistemas Operacionais: Conceitos e Aplicações**. Rio de Janeiro: LTC, 2001.

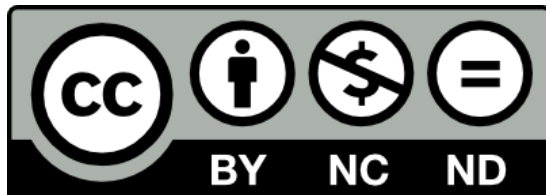


TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais: Projeto e Implementação**. 2. ed. Porto Alegre: Bookman, 2000.



MAZIERO, Carlos Alberto. **Sistemas Operacionais: Conceitos e Mecanismos [recurso eletrônico]**. Curitiba: DINF - UFPR, 2019. ISBN 978-85-7335-340-2.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Sistemas Operacionais

Comunicação Interprocessos
Programação Concorrente