



e-Learning



Research



www.professorlima.com

Aula Anterior

Escalonamento de
Processos
 SJF-Preemptivo e Não-
Preemptivo
 Round Robin rou
Alternância Circular

Aula de Hoje

Comunicação e
Sincronização de Processos
 Programação
Concorrente
 Problemas de
Compartilhamento de
Recursos

Próxima Aula

Comunicação e
Sincronização de Processos
 Programação
Concorrente
 Problemas de
Compartilhamento de
Recursos

Cronograma

Comunicação de Processos

- A comunicação entre processos ou comunicação inter-processo (IPC ou Inter Process Communication) é uma situação comum dentro dos sistemas computacionais que ocorre quando dois ou mais processos precisam se comunicar, isto é, quando os processos devem compartilhar ou trocar dados entre si.

Comunicação de Processos

- A comunicação entre processos pode ocorrer em várias situações diferentes tais como:
- redirecionamento da saída (dos resultados) de um comando para outro,
- envio de arquivos para impressão,
- transmissão de dados através da rede,
- transferência de dados entre periféricos, etc.

Comunicação de Processos

- Tal comunicação se dá, geralmente, através da utilização de recursos comuns aos processos envolvidos na própria comunicação. Como não é razoável que tal comunicação envolva mecanismos de interrupção devido à sua complexidade e limitações de performance, as interrupções são reservadas para a administração do sistema em si.
- Para a comunicação inter-processo é necessário algum mecanismo bem estruturado.

- Veremos alguns mecanismos possíveis para a comunicação de processos destacando-se:
- Buffers
- Semáforos
- Memória Compartilhada

Buffers e operações de sleep e wakeup

- Um problema típico é o do produtor-consumidor, onde dois processos distintos compartilham um buffer, uma área de dados de tamanho fixo que se comporta como um reservatório temporário.
- O processo produtor coloca informações no buffer enquanto o processo consumidor as retira de lá.

Buffers e operações de sleep e wakeup

- Se o produtor e o consumidor são processos sequenciais, a solução do problema é simples, mas caso sejam processos paralelos passa a existir uma situação de concorrência.
- Mesmo nos casos onde existam múltiplos produtores ou múltiplos consumidores o problema encontrado é basicamente o mesmo.
- Este é um problema clássico de comunicação inter-processo, tais como os problemas do jantar dos filósofos e do barbeiro dorminhoco.

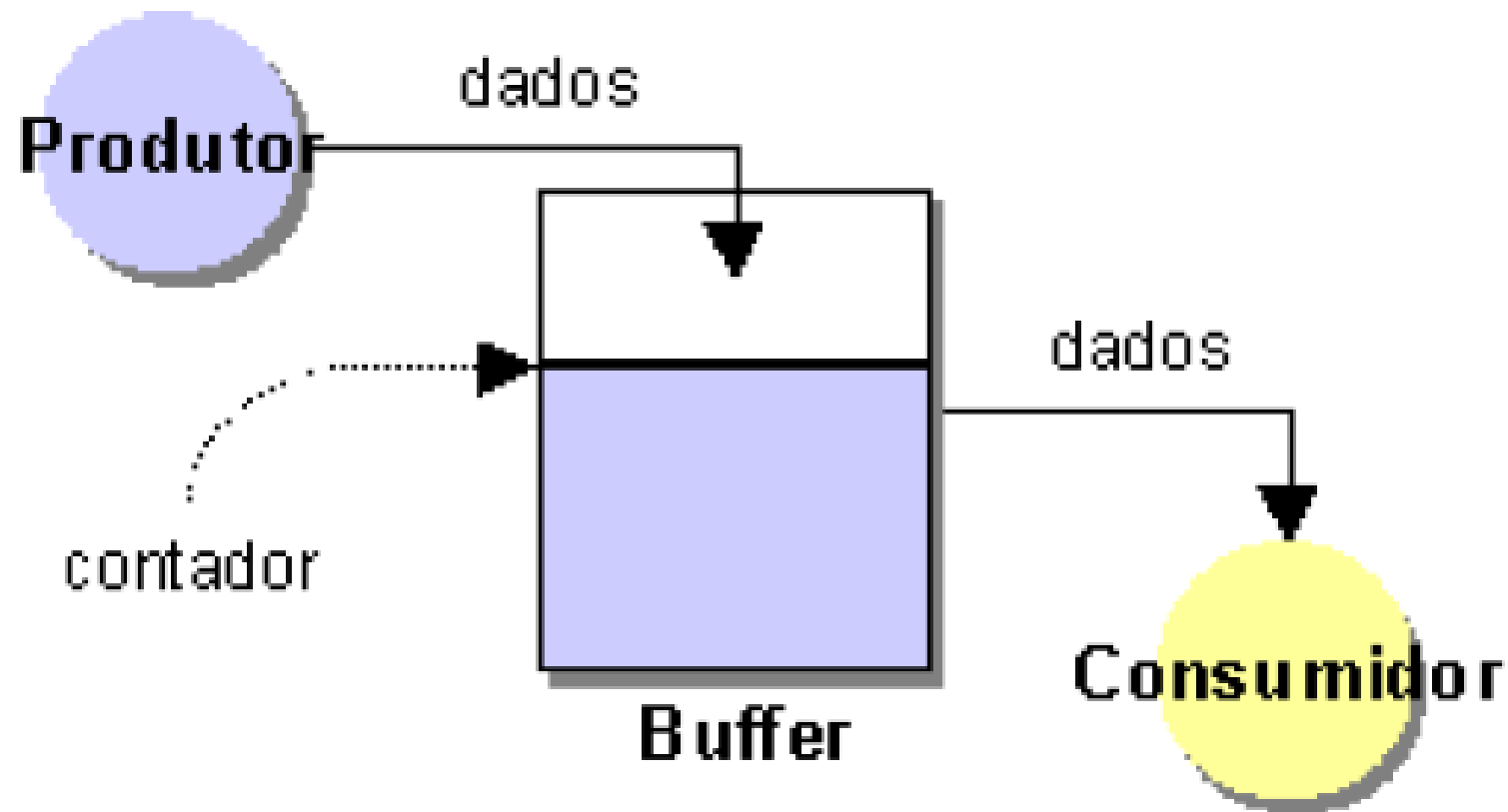
Buffers e operações de sleep e wakeup

- Programas que desejam imprimir podem colocar suas entradas (nomes dos arquivos a serem impressos ou os arquivos de impressão propriamente ditos) em uma área de spooling, denominada de printer spool.
- Um outro processo (tipicamente um daemon de impressão) verifica continuamente a entrada de entradas no spool, direcionando-as para uma ou mais impressoras existentes quando estas se tornam ociosas, com isto retirando as entradas da área de spool.
- É claro que a área reservada para o spool é finita e que as velocidades dos diversos produtores (programas que desejam imprimir) pode ser substancialmente diferente das velocidades dos consumidores (das diferentes impressoras instaladas no sistema).

Buffers e operações de sleep e wakeup

- A mesma situação pode ocorrer quando diversos processos utilizam uma placa de rede para efetuar a transmissão de dados para outros computadores.
- Os processos são os produtores, enquanto o hardware da placa e seu código representam o consumidor.
- Em função do tipo de rede e do tráfego, temos uma forte limitação na forma que a placa consegue consumir (transmitir) os dados produzidos pelos programas e colocados no buffer de transmissão.

Problema do produtor-consumidor



o produtor não pode colocar novas informações no buffer porque ele já está cheio; ou o consumidor não pode retirar informações do buffer porque ele está vazio.

- Nestes casos tanto o produtor como o consumidor poderiam ser adormecidos, isto é, ter sua execução suspensa, até que existisse espaço no buffer para que o produtor coloque novos dados ou existam dados no buffer para o consumidor possa retirá-los

Solução parcial do problema produtor-consumidor

- O produtor continuará produzindo.
- O buffer ficará cheio pois consumidor está inativo, fazendo que o produtor se coloque como inativo com outro sleep.
- Ambos os processos permanecerão para sempre inativos.

Semáforos

- Para resolver o problema produtor-consumidor, Dijkstra propôs também em 1965 a utilização de variáveis inteiras para controlar o número de sinais wakeup para uso futuro.
- Estas variáveis foram denominadas semáforos, e sobre elas estabeleceu-se duas diferentes operações: P (conhecida também como Down) e V (conhecida também como Up)

“Pensamentos valem e vivem pela observação exata ou nova, pela reflexão aguda ou profunda; não menos querem a originalidade, a simplicidade e a graça do dizer.”

Machado de Assis



Obrigado!

