

[◀ VOLTAR](#)

Sincronização de Processos

Neste tópico, iremos abordar a Sincronização de Processos, sua definição e conceito.

NESTE TÓPICO

- Introdução
- Referências



Marcar
tópico

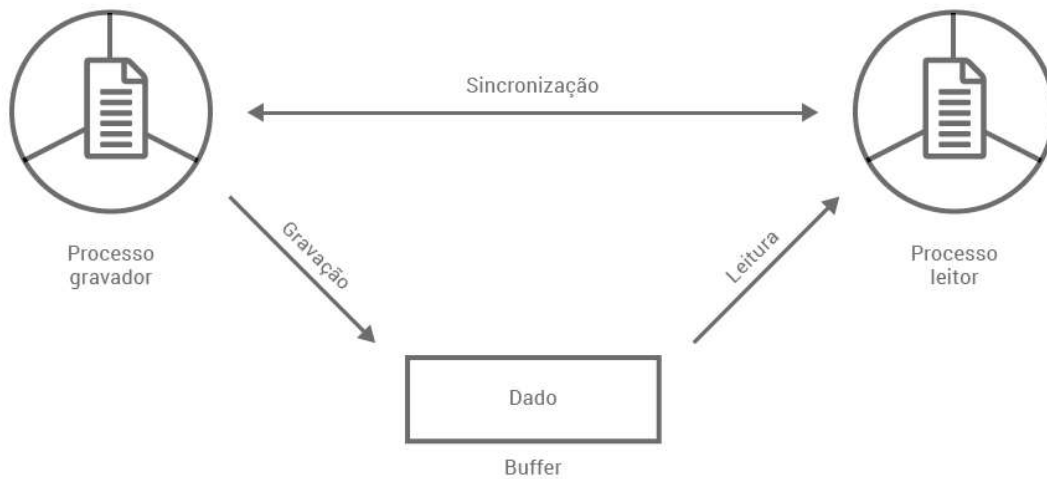


Introdução

O núcleo do sistema operacional, o Kernel pode atender várias solicitações (Kernel multithread), para tal, alguns mecanismos de sincronização devem ser utilizados para proteger suas estruturas de dados internas.

Em ambientes distribuídos, threads são essenciais para solicitações de serviços remotos. Em um ambiente monothread, se uma aplicação solicita um serviço remoto, ela pode ficar esperando indefinidamente, enquanto aguarda pelo resultado. Em um ambiente multithread, um thread pode solicitar o serviço remoto, enquanto a aplicação pode continuar realizando outras atividades úteis. Já para o processo que atende a solicitação, múltiplos threads permitem que diversos pedidos sejam atendidos concorrentemente e/ou simultaneamente. Essas situações exigem o sincronismo entre processos.

A figura abaixo ilustra uma sincronização entre dois processos que utilizam o Buffer:



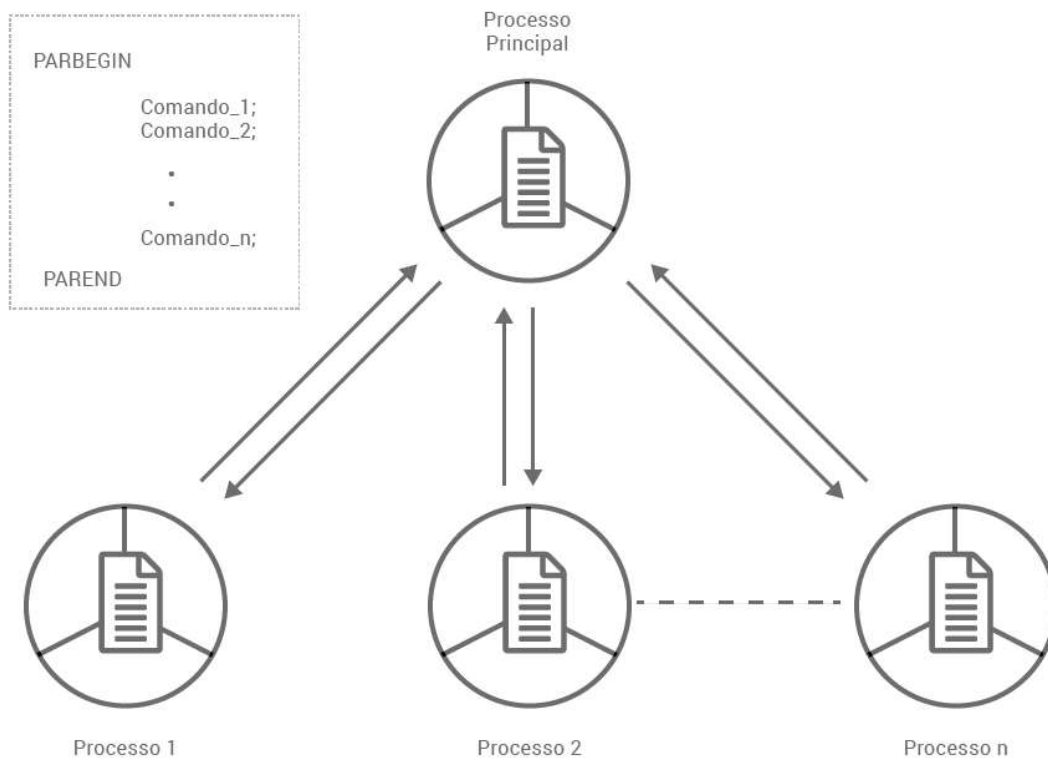
Sincronismo entre dois Processos.

Fonte: MACHADO, F. B.; MAIA, L. P. Arquitetura de sistemas operacionais. 2. ed. São Paulo: LTC, 2002.

Importante:

Buffer: é uma área reservada na RAM (memória física) utilizada para armazenar dados temporariamente. O Buffer é sempre utilizado para otimizar os processos, liberando recursos como o processador.

Um outro exemplo de sincronismo é mostrado na figura abaixo:



Sincronismo entre três processos com o Processo Principal.

Fonte: MACHADO, F. B.; MAIA, L. P. Arquitetura de sistemas operacionais. 2. ed. São Paulo: LTC, 2002.

Na figura acima, notamos o sincronismo de atividades entre cada um dos 3 processos com o processo principal. O sincronismo ocorre para que não ocorra um deadlock (trava no Sistema) entre os processos envolvidos no processamento.

Pthreads

Uma das grandes dificuldades da utilização de threads em aplicações em geral foi a ausência de um padrão. Em 1995, o padrão POSIX (Portable Operating System Interface) P1003.1c foi aprovado. Com este padrão, também conhecido como Pthreads, aplicações comerciais multithread tornaram-se mais comuns. O padrão Pthreads é largamente encontrado em ambientes Unix, geralmente implementado em aplicações escritas em Linguagem C.

No padrão POSIX.1c, threads são criados e eliminados dinamicamente, conforme a necessidade. Além disto, o padrão oferece mecanismos de sincronização, como semáforos, mutexes e variáveis condicionais. O Pthreads oferece inúmeras funções para a sincronização entre threads. A seguir descrevemos, resumidamente, estes mecanismos:

- **Mutexes:** Uma variável mutex (Mutual Exclusion) permite implementar a exclusão mútua, ou seja, o acesso a uma região crítica é feito de forma a impedir os problemas de sincronização (race conditions);
- **Variáveis condicionais:** Uma variável condicional permite associar uma variável a um evento, que poder ser, por exemplo, um contador alcançando um determinado valor ou um flag sendo ligado/desligado. Em uma aplicação multithread, a utilização de variáveis condicionais permite criar um mecanismo de comunicação entre os diversos threads, que, por sua vez, permite a execução condicional em torno de eventos (sincronização);
- **Semáforos:** Semáforos são semelhantes aos mutexes, com a diferença que, enquanto um mutex pode variar seu valor entre zero e um, semáforos podem assumir outros valores, permitindo sua utilização como contadores. Semáforos estão disponíveis apenas em sistemas que oferecem suporte ao padrão POSIX com extensões para tempo-real (POSIX.1b).

Há uma grande dificuldade em se determinar a ordem de ocorrência de um evento. Os atrasos de comunicação, em uma rede distribuída, são imprevisíveis, podem ocorrer em qualquer momento e em várias situações. Dessa forma foram definidas ordens para as sincronizações:

▪ Ordenação causal

- Assegura que todos os processos reconheçam que um evento causalmente dependente deve ocorrer apenas após o evento do qual é dependente;

- É implementada pela relação acontece-antes:
 - Se os eventos a e b pertencerem ao mesmo processo, então $a \rightarrow b$ se a ocorrer antes de b ;
 - Se o evento a é o emissor de uma mensagem e o evento b é o receptor da mensagem, então $a \rightarrow b$;
 - Essa relação é transitiva.
 - Apenas uma ordenação parcial:
 - Os eventos em relação aos quais não é possível determinar qual ocorreu primeiro são considerados concorrentes.
-
- **Ordenação total**
 - Possibilita que todos os eventos sejam ordenados e que a causalidade seja preservada;
 - Pode ser implementada por meio de um relógio lógico que atribui uma marca de tempo (*timestamp*) a cada evento que ocorre no sistema;
 - Os relógios lógicos escalares sincronizam os relógios lógicos nos hospedeiros remotos e garantem a causalidade.
-
- **Vários métodos de sincronização são implementados para impor a exclusão mútua nos sistemas distribuídos**
 - Passagem de mensagem;
 - Algoritmo de exclusão mútua distribuída de Agrawala e Ricart.

O sincronismo entre processos é uma providência que otimiza as execuções nos computadores. A distribuição e recebimento de retorno de informações nos processamentos, sob uma determinada ordem (sincronizadas) corroboram com a melhor maneira de serem processadas as tarefas (rotinas).

Este tópico finaliza aqui, esperamos que você tenha entendido a importância do sincronismo nas execuções dos processos.

Bom estudo!

Quiz

Exercício

Sincronização de Processos

INICIAR >

Quiz

Exercício Final

Sincronização de Processos

INICIAR >

Referências

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

MACHADO, F. B.; MAIA, L. P. Arquitetura de sistemas operacionais. 2. ed. São Paulo: LTC, 2002.

TANENBAUM, A. S. Sistemas operacionais modernos. 2. ed. São Paulo: Pearson Prentice Hall, 2005.



Avalie este tópico



ANTERIOR
Deadlock

Índice

Biblioteca
(https://www.uninove.br/conheca-
a-
uninove/biblioteca/sobre-
a-
biblioteca/apresentacao/)
Portal Uninove
(http://www.uninove.br)
Mapa do Site

Ajuda?
(https://ava.un
idCurso=)
Gerenciamento de Mem
© Todos os direitos reservados