Sincronização de processos

Passagem de mensagens

ARILSON FERREIRA DOS SANTOS
CLAUDIO ALEX MESSIAS DA ROSA
EDUARDO LUIZ SCHADE SOARES
SILVIA TATYARA RODRIGUES LOPES

Problema:

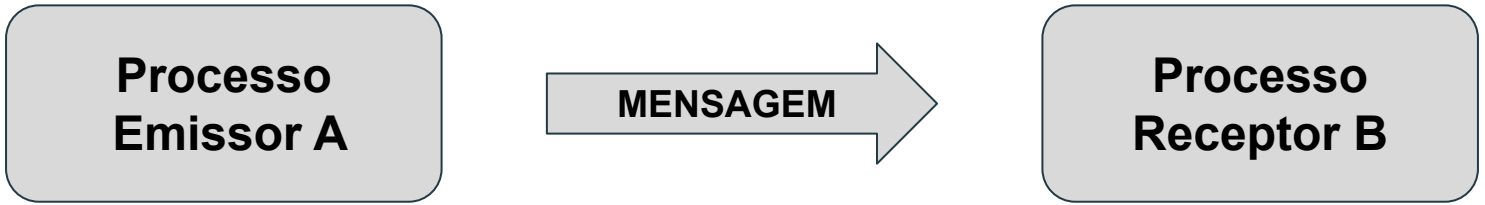
Os semáforos e algoritmos de exclusão mútua são baseados no compartilhamento de variáveis. Isso implica na necessidade de compartilhamento de memória física, mas em sistemas distribuídos, às máquinas formam unidades independentes os então chamados “nós”, então não há memória compartilhada entre os processos.

Solução : Esquema de troca de mensagens:

Os processos enviam e recebem mensagens, em vez de ler e escrever em variáveis compartilhadas.

Sincronização entre processos: Garantida pela restrição de que uma mensagem só poderá ser recebida depois de ter sido enviada.

A transferência de dados de um processo para outro, após ter sido realizada a sincronização, estabelece a comunicação.



As primitivas podem ser de dois tipos:

Bloqueantes(Blocking): quando o processo que as executar ficar bloqueado até que a operação seja bem sucedida. Quando ocorrer a entrega efetiva da mensagem ao processo destino, no caso da emissão. Quando ocorrer o recebimento da mensagem pelo processo destino, no caso de recepção).

Não bloqueantes (Nonblocking): quando o processo que executar a primitiva continua sua execução normal, independentemente da entrega ou do recebimento efetivo da mensagem pelo processo destino.

Combinação de Primitivas :

Envia Bloqueante – Recebe Bloqueante → Síncrono

Envia Bloqueante – Recebe Não Bloqueante

Envia Não Bloqueante – Recebe Bloqueante → Semi-síncrono

Envia Não Bloqueante – Recebe Não Bloqueante → Assíncrono

Primitivas de Endereçamento Direto

- Este mecanismo permite uma comunicação simultânea entre os processos, permitindo uma maior sincronização e comunicação entre os mesmos.
- No caso de a comunicação entre os processos ser feitas através de mensagens, o produtor é o **Emissor** e o consumidor é o **Receptor** da mensagem.

Primitivas de Endereçamento Direto	
Send	(receptor, mensagem)
Receive	(emissor, mensagem)

Primitivas de Endereçamento indireto

- Através deste método, os processos não realizam uma comunicação sincronizada, mas em forma de **mailbox** que residem no núcleo do SO.
- A mailbox é uma fila de mensagens onde por meio delas, podem haver inúmeros processos receptores.
- Por se tratar de uma fila, a mailbox segue uma linguagem FIFO (First in First out), mas caso uma mensagem precise ser enviada urgente neste caso é criada uma fila de prioridade.

Mecanismos de comunicação entre processos:

- **RPC – Remote Procedure Call** : Rotinas que permitem comunicação de processos em diferentes máquinas; ◦ Chamadas remotas. Onde um processo pode comandar a execução de um procedimento situado em outra máquina , o processo chamador deverá ficar bloqueado até o procedimento chamado termine tanto a chamada quanto o retorno podem envolver troca de mensagem passando parâmetros

- **MPI – Message-passing Interface**: São sistemas paralelos

- **RMI Java – Remote Method Invocation**: Permite que um objeto ativo em uma máquina virtual Java possa interagir com objetos de outras máquinas virtuais Java, independentemente da localização dessas máquinas virtuais;

- **Web Services**: Permite que serviços sejam compartilhados através da Web.

Mecanismos de comunicação entre processos:

- **Pipe**: Permite a criação de filas de processos, ps -ef | grep produtor_consumidor; ◦ Saída de um processo é a entrada de outro; ◦ Existe enquanto o processo existir;
- **Named pipe**: Extensão de pipe ◦ Continua existindo mesmo depois que o processo terminar; ◦ Criado com chamadas de sistemas;
- **Socket**: Par endereço IP e porta utilizado para comunicação entre processos em máquinas diferentes; ◦ Host X (192.168.1.1:1065) Server Y (192.168.1.2:80);

Problemas do Sistema de Troca de Mensagem

Os sistemas de troca de mensagens possuem alguns problemas e estudos de projetos interessantes Principalmente quando os processos comunicantes estão em máquinas diferentes, conectadas por uma rede de comunicação. Os principais são:

Perda de mensagens

Perda de reconhecimento

Nomeação de Processos

Autenticação

Estudos de Projeto para quando emissor e receptor estiverem na mesma máquina

Solução Perda de Mensagem

Possível solução: O receptor, ao receber uma nova mensagem, envia uma mensagem especial de reconhecimento (ACK).

Se o emissor não receber um ACK dentro de um determinado intervalo de tempo, deve retransmitir a mensagem.

Problema da transação bancária

Um usuário deseja realizar duas transações ao mesmo tempo, em duas contas diferentes:

deb (conta1, conta2);

cred(conta2, conta1).

Debitar um alto valor de uma conta e creditar o mesmo valor na outra conta.

(LEITE, 2009).

Problema da transação bancária

- Atomicidade: realizar as transações de forma indivisível;
- Consistência: sem violar as regras do sistema;
- Isolamento: transações concorrentes não devem acontecer;
- Durabilidade: manter alterações após o commit.

(PRETTI, 2005).

Problema da transação bancária

Como saber se as duas transações **$$$** foram realizadas com falha de rede após a primeira movimentação?

Atomicidade

Realizar todas as transações ou nãorealizar nenhuma transação.

Referências:

http://www.univasf.edu.br/~andreza.leite/aulas/SO/ProcessosSincronizacao.pdf

http://inf.ufes.br/~zegonc/material/Sistemas_Operacionais/Sincronizacao(5)-Troca%20de%20Mensagens.pdf

http://www.di.ubi.pt/~operativos/teoricos/capitulo7.pdf

http://wiki.icmc.usp.br/images/7/76/Aula06.pdf

Leite, Andreza. "Sistemas operacionais." Florianópolis: IF-SC (2009).

PRETTI, Marco. A message-passing algorithm with damping. Journal of Statistical Mechanics: Theory and Experiment, v. 2005, n. 11, p. P11008, 2005.