Universidade Federal do Rio de Janeiro
Informática DCC/IM

Arquitetura de Computadores II

Sistemas de Troca de Mensagens

Gabriel P. Silva

Sistemas de Comunicação

- O Sistema de Comunicação provê tipicamente os seguintes serviços para as aplicações:
  - transmissão da mensagem do processador origem ao processador destino;
  - garantia de que as mensagens entre um par de processadores origem e destino serão entregues na ordem em que foram enviadas;

Sistemas de Comunicação

- garantia de não ocorrência de *"deadlocks"*;
- confiabilidade na transmissão das mensagens.
- É necessário então que o sistema de comunicação tenha uma *camada de software* para prover, através de protocolos implementados em software, os serviços necessários a partir dos recursos de hardware disponíveis.

Comunicação Síncrona

- O processo que deseja enviar uma mensagem faz uma requisição inicial de autorização para enviar a mensagem ao processo destinatário.
- Depois de receber essa autorização, a operação de envio (*"send"*) é realizada.
- A operação de envio só se completa quando a operação de recepção (*"receive"*) correspondente retorna uma mensagem de *"acknowlegement"*.
- Portanto, ao se concluir a operação de envio, os *buffers* e estruturas de dados utilizados na comunicação podem ser reutilizados.

Comunicação Síncrona

- Este modo de comunicação é simples e seguro, contudo elimina a possibilidade de haver superposição entre o processamento da aplicação e o processamento da transmissão das mensagens.
- Requer um protocolo de quatro fases para a sua implementação do lado do transmissor: pedido de autorização para transmitir; recebimento da autorização; transmissão propriamente dita da mensagem; e recebimento da mensagem de *"acknowledgement"*.

Comunicação Assíncrona

- Bloqueante
  - A operação de envio (*"send"*) só retorna o controle para o processo que a chamou após ter sido feita a cópia da mensagem a ser enviada de um *buffer* da aplicação para um *buffer* do sistema operacional.
  - Portanto, ao haver o retorno da operação de envio, a aplicação está livre para reutilizar o seu *buffer*, embora não haja nenhuma garantia de que a transmissão da mensagem tenha completado ou vá completar satisfatoriamente.

Comunicação Assíncrona

- A operação de envio bloqueante difere da operação de envio síncrona, uma vez que não é implementado o protocolo de quatro fases entre os processos origem e destino.
- A operação de recepção (*"receive"*) bloqueante é semelhante à operação de recepção síncrona, só retornando para o processo que a chamou após ter concluído a transferência da mensagem do *buffer* do sistema para o *buffer* especificado pela aplicação. A diferença em relação a operação de recepção síncrona é que a mensagem de *"acknowledgement"* não é enviada.

Comunicação Assíncrona

- Não Bloqueante
  - Na primeira fase, a operação de *"send"* retorna imediatamente, tendo apenas dado início ao processo de transmissão da mensagem e a operação de *"receive"* retorna após notificar a intenção do processo de receber uma mensagem.
  - As operações de envio e recepção propriamente ditas são realizadas de forma assíncrona pelo sistema.
  - O retorno das operações de *"send"* e de *"receive"* nada garantem e não autoriza a reutilização das estruturas de dados para nova mensagem.

Comunicação Assíncrona

- Na segunda fase, é verificado se a operação de troca de mensagens iniciada anteriormente pelas primitivas *"send"* e*"receive"* já foi concluída.
- Somente após esta verificação é que o processo que realizou o envio de dados pode alterar com segurança sua área de dados original.
- Este modo de comunicação é o que permite maior superposição no tempo entre computações da aplicação e o processamento da transmissão das mensagens.

Arquitetura de Software dos Sistemas de Comunicação

- Cada mensagem percorre o seguinte caminho:
  - nó de processamento origem ➔
  - interface de rede do nó de origem ➔
  - rede de interconexão ➔
  - interface de rede do nó destino ➔
  - nó de processamento destino.
- A interação do nó de processamento com a sua interface de rede é realizada através de chamadas ao sistema operacional.

Arquitetura de Software dos Sistemas de Comunicação

- A latência total na comunicação por troca de mensagens ser definida pela soma das seguintes parcelas:

  latência total = overhead de software + overhead da interface da rede + latência da rede de interconexão

- O *"overhead"* de software é devido a um conjunto de fatores:
  - complexidade do protocolo de comunicação,
  - processamento de interrupções
  - Troca de contexto devido às chamadas ao sistema
  - cópias dos dados entre *"buffers"*, etc.

Arquitetura de Software dos Sistemas de Comunicação

- As técnicas utilizadas para diminuir este *"overhead"*:
  - Utilização de protocolos mais simples, seja através da simplificação dos serviços oferecidos ou através do uso de hardware capaz de implementar diretamente alguns dos serviços;
  - Eliminação da necessidade de se realizar troca de contexto, permitindo a implementação da comunicação no modo usuário;
  - Redução ou a eliminação das cópias dos dados, permitindo a manipulação deles diretamente em espaço de memória do usuário.

Arquitetura de Software dos Sistemas de Comunicação

- A definição da responsabilidade pela execução das funções de comunicação pode ter grande impacto no desempenho do sistema. Com alternativa, possibilita-se o acesso direto da aplicação do usuário à interface de rede (abordagens mais recentes).

Arquitetura de Software dos Sistemas de Comunicação

- Exemplos deste tipo de arquitetura os sistemas
  - U-Net da Universidade de Cornell
  - Shrimp da Universidade de Princeton
  - Hamlyn da Hewllet-Packard
  - Memory Channel da Digital
  - *"Virtual Interface Architecture"* - VIA proposta pela Compaq, Microsoft e Intel.

Arquitetura de Software dos Sistemas de Comunicação

- **Envio dos dados:**
  - Através de E/S programada.
  - Por acesso direto à memória (DMA – *"Direct Memory Access"*).
    - Mensagens longas
    - Controladores de DMA usam endereços físicos e a aplicação conhece apenas endereços virtuais
  - Soluções:
    - Atribuir ao sistema operacional o comando da operação de DMA → mensagem é copiada para um buffer do núcleo
    - Escrita Remota em Memória

Arquitetura de Software dos Sistemas de Comunicação

- **Escrita Remota em Memória**
  - Transfere a mensagem para uma região de memória no espaço do usuário, com endereço físico conhecido, constituída de páginas marcadas como não removíveis (*"unswappable"*) da memória
  - A essa região dá-se, em geral, o nome de *"pinned-down buffer"*
  - Uma variação deste esquema pode utilizar uma estrutura do tipo TLB, preenchida pelo sistema operacional, para conversão de endereços virtuais em físicos pela própria aplicação. As páginas mapeadas nesta TLB específica são marcadas como não removíveis
  - Utilizado no SHRIMP e VIA

Arquitetura de Software dos Sistemas de Comunicação

- **Notificação de chegada de mensagem:**
  - **Interrupção:**
    - **O *"overhead"* para tratamento de interrupções pode ser considerável.**
  - ***"Polling"* da interface da rede:**
    - **Não há o overhead do caso anterior**
    - **Se a freqüência da operação de *"polling"* for muito baixa, a latência para recepção de uma mensagem pode aumentar desnecessariamente.**
    - **Se a freqüência for muito alta, pode ser re-introduzido um *"overhead"* no processador com várias verificações desnecessárias da chegada de cada mensagem.**

Arquitetura de Software dos Sistemas de Comunicação

- **Esquema alternativo:**
  - **Baseado no uso de mensagens que levam o endereço de uma rotina no nível do usuário que deve ser executada no receptor quando de sua chegada.**
  - **O *"handler"* extrai a mensagem da rede e a integra à computação em andamento no receptor.**
    - ***Active Messages* da Universidade de Berkeley**
    - ***Fast Messages* da Universidade de Illinois**