

Troca de Mensagens

Tópicos:

- Características de Mensagens
- Implementação de Troca de Mensagens
- Padronizações: PVM, MPI

Características de Mensagens

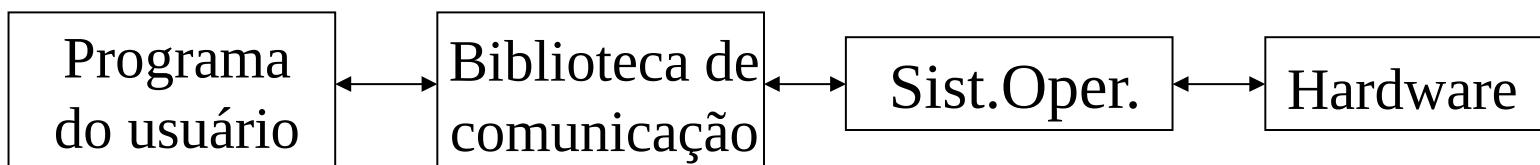
- Características típicas de uma *mensagem*:
 - Contém um conjunto de ítems do mesmo tipo
 - Na memória, ítems podem estar contíguos ou não
 - Ítems são recebidos na mesma ordem em que são transmitidos (garantido por hardware ou software)
- Principais parâmetros de uma mensagem:
 - Tipo de cada ítem
 - Endereço do primeiro ítem
 - Intervalo entre ítems
 - Número de ítems (comprimento da mensagem)
 - Processador(es) de destino

Características de Mensagens (cont.)

- Semântica na troca de mensagens:
 - *send/receive* podem ser síncronas ou assíncronas
 - Caso síncrono: processadores de envio e de recebimento efetuam a comunicação simultaneamente; primeiro a chegar espera pelo outro (pouco usado atualmente)
 - Caso assíncrono: processador de envio pode efetuar a comunicação sem esperar pelo processador de recebimento; dados em trânsito devem ser armazenados de alguma forma
 - *send/receive* podem ser com ou sem bloqueio
 - Caso com bloqueio: só há um retorno das funções *send/receive* quando o processador já terminou sua participação na comunicação
 - Caso sem bloqueio: pode haver um retorno da função *send/receive* sem que a comunicação tenha ocorrido ainda

Implementação de Troca de Mensagens

- Implementação prática típica:
 - Funções de *send/receive*
 - Biblioteca com funções de comunicação, fornecida pelo próprio fabricante do sistema
 - Interfaces para linguagens tradicionais (C, Fortran, etc)
 - Funções da biblioteca acessam internamente o Sistema Operacional



Implementação de Troca de Mensagens (cont.)

- Bibliotecas de comunicação:
 - Fornecidas junto com o sistema/hardware
 - Otimizadas pelo fabricante para o seu sistema
 - Em geral, suportam estritamente as funcionalidades disponíveis no hardware
 - Conseqüências:
 - Programas finais são pouco portáveis entre sistemas
 - Mesmo entre modelos diferentes do mesmo fabricante, pode ser necessário reescrever um código-fonte para otimizá-lo
 - Mudança de sistema requer reaprendizado dos programadores

Padronizações

- Objetivo básico:
 - Criar biblioteca uniforme, disponível em todos os sistemas
- Vantagens:
 - Portabilidade ampla para os programas
- Desvantagem:
 - Desempenho num certo sistema pode ser não-ótimo
- Principais Iniciativas:
 - PVM (Parallel Virtual Machine)
 - MPI (Message Passing Interface)

Parallel Virtual Machine (PVM)

- Estratégia:
 - Proposto pelo Oak-Ridge Nat. Lab (EUA): \approx 1990
 - Objetivo original: uso em redes de estações de trabalho
 - Implementação: criar um ambiente paralelo “virtual”
- Situação Atual:
 - Várias versões já implementadas
 - Disponível em vários sistemas paralelos
 - Suporte de alguns fabricantes \rightarrow bom desempenho
 - Versão de domínio público disponível

Message Passing Interface (MPI)

- Estratégia:
 - Especificada por um fórum: mais de 40 instituições
 - Representantes de Universidade/Indústria/Inst.Pesquisa
 - Participação inclusive dos criadores de PVM
- Situação Atual:
 - Três versões do padrão já publicadas: 1.0, 1.1, 2.0
 - Disponível em quase todos os sistemas paralelos
 - Suporte de vários fabricantes → bom desempenho
 - Várias versões de domínio público disponíveis