
Sistemas Distribuídos

— Comunicação em SD —

Questões

- Porque os tradicionais mecanismos de comunicação entre processos utilizados em sistemas centralizados não podem ser utilizados em sistemas distribuídos? Como é possível permitir a comunicação entre processos em um sistema distribuído?

Introdução: Comunicação em Sistemas Distribuídos

- A diferença mais importante entre Sistemas Distribuídos e Sistemas Centralizados é a Comunicação Interprocesso (Interprocess Communication – IPC);
- Sistemas Centralizados
 - memória compartilhada;
- Sistemas Distribuídos
 - não existe memória compartilhada;
 - Comunicação Interprocesso reformulada
 - Permitir que os processos se comuniquem para troca de dados ou acessos a recursos ou serviços em processadores remotos.

Introdução: Comunicação Interprocessos

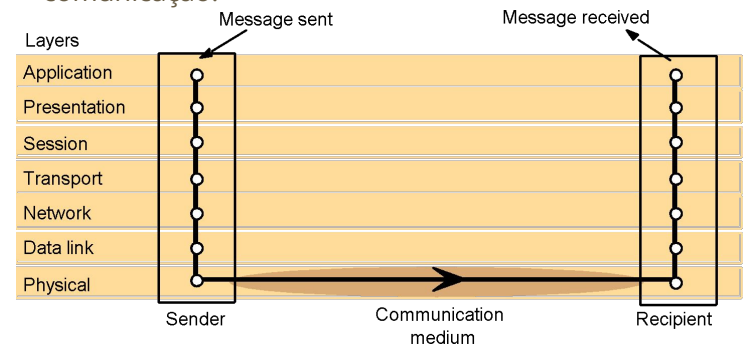
- Sistema Distribuído é baseado na Troca de Mensagens (Message Passing);
- Exemplo:
 - Quando um Processo-A quer se comunicar com um Processo-B, o Processo-A constrói uma mensagem no seu próprio espaço de endereçamento (address space);
 - Em seguida, o Processo-A executa uma chamada de sistema (system call) que faz com que o sistema operacional pegue essa mensagem e envie pela rede para o Processo-B;
- Quais são os desafios (acordos e regras) necessários que permitem que os Processos A e B se comuniquem?

Protocolos: Conceitos Básicos

- Conjunto de regras, procedimentos e formatos para garantir a comunicação entre duas entidades geograficamente distintas
 - A sequência de mensagens que devem ser trocadas;
 - O formato dos dados nas mensagens.
- ISO (International Standard Organization), desenvolveu um modelo de referência
 - diversos níveis envolvidos
 - o que deve ser tratado em cada nível;
 - Open Systems Interconnection Reference Model, ISO OSI ou, simplesmente, Modelo OSI.

Protocolos: Camadas

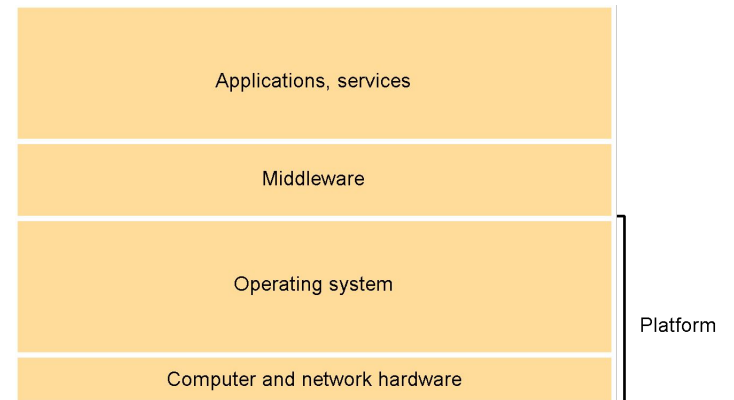
- Comunicação é dividida em sete níveis ou camadas, onde cada camada trata de um aspecto específico da comunicação.



Protocolos: Desafios

- Roteamento:
 - Prover o caminho mais eficiente para um pacote, através da aplicação de algoritmos de roteamento
- Controle de Congestionamento:
 - Evitar a degradação na vazão da rede através de atrasos no envio de pacotes;
 - Informar aos participantes da rota do pacote sobre o congestionamento.
- Internetworking:
 - Integrar diversos tipos de redes, endereçamento, protocolos, componentes de ligação (roteadores, bridges, hubs, switches).

Comunicação Interprocessos: Camadas de Serviços

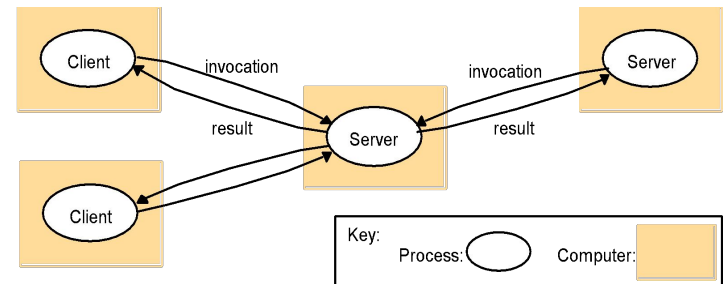


Comunicação Interprocessos: Modelo Cliente-Servidor

- A ideia é estruturar o sistema operacional como um grupo de processos cooperativos chamados:
 - Servidores: Oferecem serviços aos usuários;
 - Clientes: Usam os serviços provido pelos Servidores.
- Uma máquina pode executar:
 - Um único ou múltiplos processos clientes;
 - Um único ou múltiplos processos servidores;
 - Ou uma combinação das alternativas anteriores.

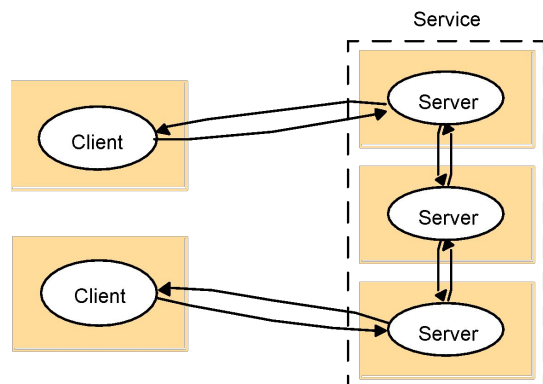
Comunicação Interprocessos: Modelo Cliente-Servidor

- Processos Clientes acionando individualmente Processos Servidores:



Comunicação Interprocessos: Modelo Cliente-Servidor

- Um serviço provido por Múltiplos Processos Servidores:

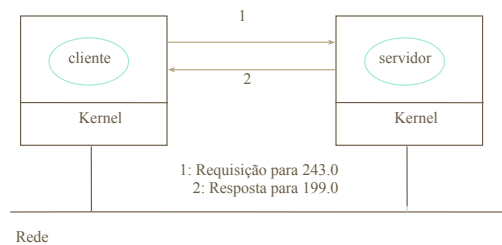


Endereçamento



Troca de Mensagem: Endereçamento

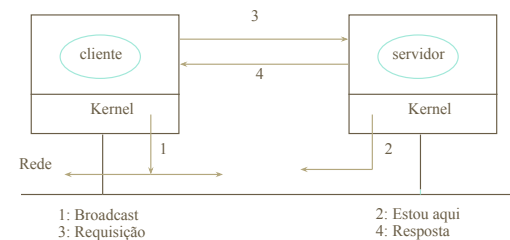
- Endereçamento Máquina Processo:



- Não é transparente
 - se um servidor não estiver disponível teremos recompilação para poder realizar o serviço em outro servidor.

Troca de Mensagem: Endereçamento

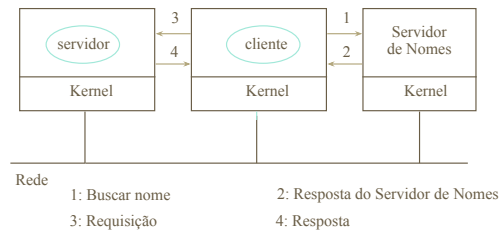
- Endereçamento Aleatório:



- Broadcast gera carga extra de comunicação no sistema.

Troca de Mensagem: Endereçamento

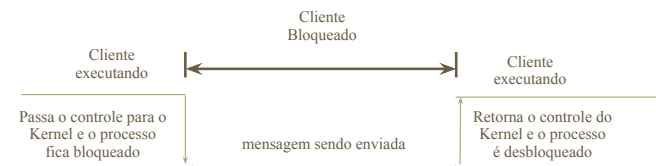
- Endereçamento usando um Servidor de Nomes:



- Problemas ????

Troca de Mensagem: Primitiva Send

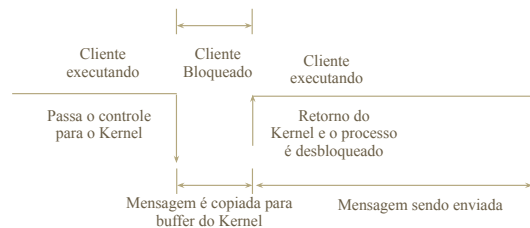
- Primitivas Bloqueadas (Síncronas):
 - Primitivas vistas até agora (send e receive) são chamadas primitivas bloqueadas.
 - Enquanto a mensagem está sendo enviada ou recebida, o processo permanece bloqueado (suspensão).



Troca de Mensagem: Primitiva Send

- Primitivas Não Bloqueadas (Assíncronas):

- Quando um send é executado o controle retorna ao processo antes da mensagem ser enviada;
- O processo que executa o send pode continuar processando enquanto a mensagem está sendo enviada.



Troca de Mensagem: Primitiva Send

- Algumas Considerações:

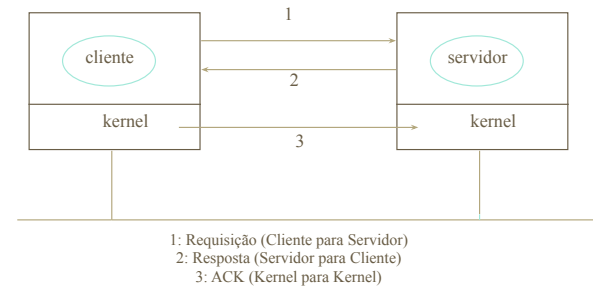
- A escolha do tipo de primitiva a ser usado (bloqueada ou não bloqueada) é feita pelo projetista do sistema;
- O processo que executa o send não pode modificar o buffer de mensagem até a mensagem ter sido enviada;
- Existem dois modos para o processo ficar sabendo que a mensagem já foi enviada:
 - Send com cópia - desperdício do tempo de CPU com a cópia extra;
 - Send com interrupção - torna a programação mais difícil e não portátil

Questões

- Como implementar no cliente mecanismos de tolerância a falhas relacionados às chamadas enviadas aos servidores?

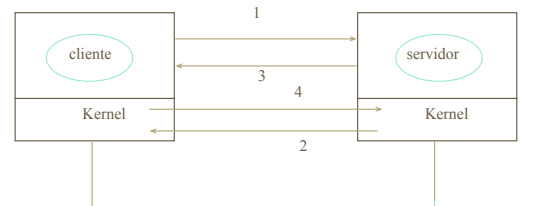
Troca de Mensagem: Confiabilidade das Primitivas

Resposta sendo usada como ACK



Troca de Mensagem: Confiabilidade das Primitivas

Mensagens de ACK individual



- 1: Requisição (Cliente para Servidor)
2: ACK (Kernel para Kernel)
3: Resposta (Servidor para Cliente)
4: ACK (Kernel para Kernel)

Troca de Mensagem: Implementação

- Detalhes de como a passagem de mensagem é implementada depende das escolhas feitas durante o projeto;
- Algumas considerações:
 - Há um tamanho máximo do pacote transmitido pela rede de comunicação;
 - Mensagens maiores precisam ser divididas em múltiplos pacotes que são enviados separadamente;
 - Alguns dos pacotes podem ser perdidos ou chegar na ordem errada;
 - Solução: Atribuir a cada mensagem o número da mensagem e um número de seqüência.

Troca de Mensagem: Implementação

- O acknowledgment pode ser para cada pacote individual ou para a mensagem como um todo;
 - No primeiro caso na perda de mensagem, somente um pacote precisa ser re-transmitido, mas na situação normal requer mais pacotes na rede de comunicação;
 - No segundo caso há a vantagem de menos pacotes na rede mas a desvantagem da recuperação no caso de perda de mensagem é mais complicada;
- Conclusão: A escolha de um dos dois métodos depende da taxa de perdas na rede.

Protocolo Cliente/Servidor

| Código | Tipo | De | Para | Significado |
|--------|-----------------|---------------------|---------------------|------------------------------------|
| REQ | Requisição | Cliente | Servidor | Solicitação de Serviço |
| REP | Resposta | Servidor | Cliente | Resposta à solicitação |
| ACK | Confirmação | Cliente Servidor | Servidor Cliente | A última mensagem chegou |
| AYA | Are You Alive? | Cliente | Servidor | Testa se o servidor está ativo |
| IAA | I Am Alive | Servidor | Cliente | Servidor ativo |
| TA | Try Again | Servidor | Cliente | Não posso atender |
| AU | Address Unknown | Servidor | Cliente | Não há processos com este endereço |

Protocolo Cliente/Servidor

Protocolo Usado na Comunicação
Cliente-Servidor (a)



Protocolo Usado na Comunicação
Cliente-Servidor (b)



Protocolo Cliente/Servidor

Protocolo Usado na Comunicação
Cliente-Servidor (c)



Protocolo Usado na Comunicação
Cliente-Servidor (d)

