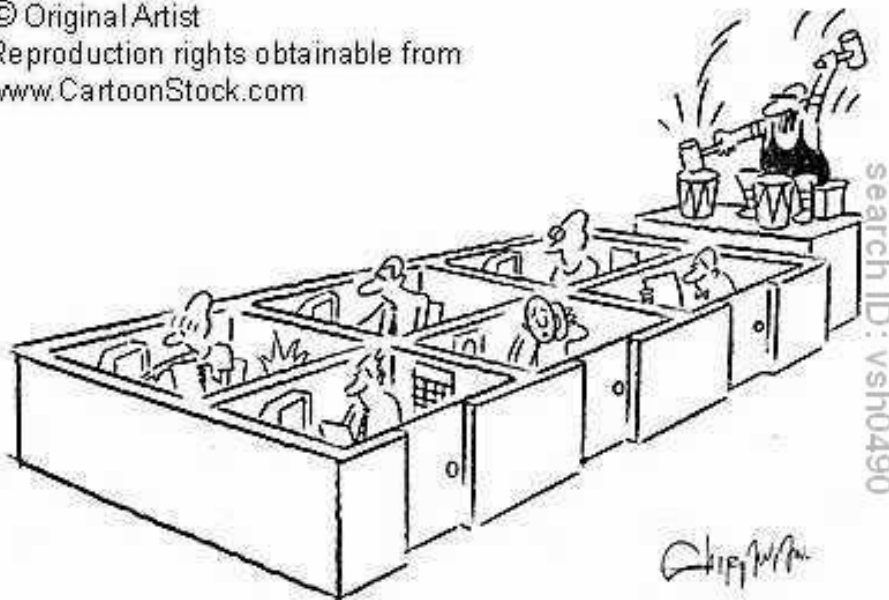


Vamos fazer um pequeno experimento

- Dividam-se em dois grupos:
 - Mestre
 - Escravo

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



I'M DISAPPOINTED IN YOU WALTER. ALL
THESE YEARS AND I FIND OUT YOU'VE
BEEN FAKING YOUR SCREAMS OF AGONY!

Projeto de Sistemas Distribuídos

Comunicação entre Processos

Prof. Msc. Marcelo Iury de Sousa Oliveira

marceloiury@gmail.com

<http://sites.google.com/site/marceloiury/>



Porque processos se comunicam?

Se comunicam porque...

- Em muitos casos os processos necessitam trocar informação de forma controlada para
 - dividir tarefas
 - aumentar da capacidade de processamento
 - sincronizar a execução de cada um.
- Além disso, possibilitar o compartilhamento de:
 - Memória;
 - Arquivos;
 - Dispositivos de E/S;
 - Etc.



pop001 www.fotosearch.com

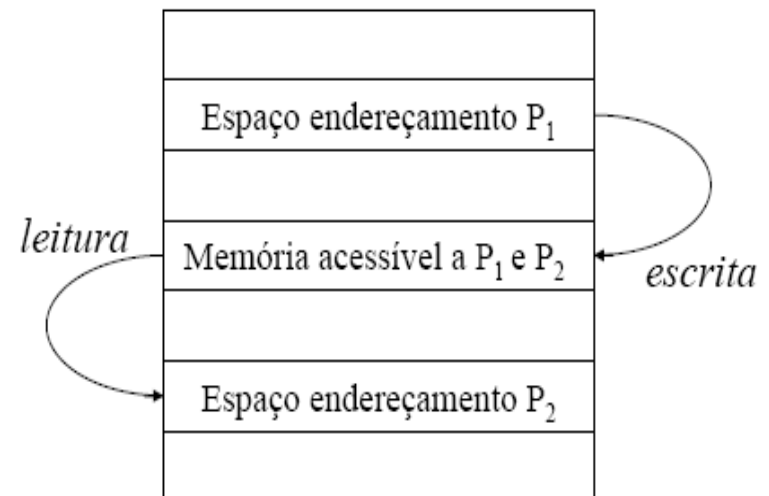
Formas de Comunicação

- Há diversas formas que possibilitam a comunicação entre processos
 - Memória Compartilhada
 - Sistema de Arquivos
 - Base de Dados
 - Chamada de Procedimento Remota
 - Middleware orientado a mensagens
 - Fluxo de dados
- O problema é escolher!!!!



Memória Compartilhada

- Os processos compartilham variáveis e trocam informações através do uso de variáveis compartilhadas.
- Vantagens:
 - Desempenho
- Desvantagens:
 - Limitação de escala
 - Sincronização

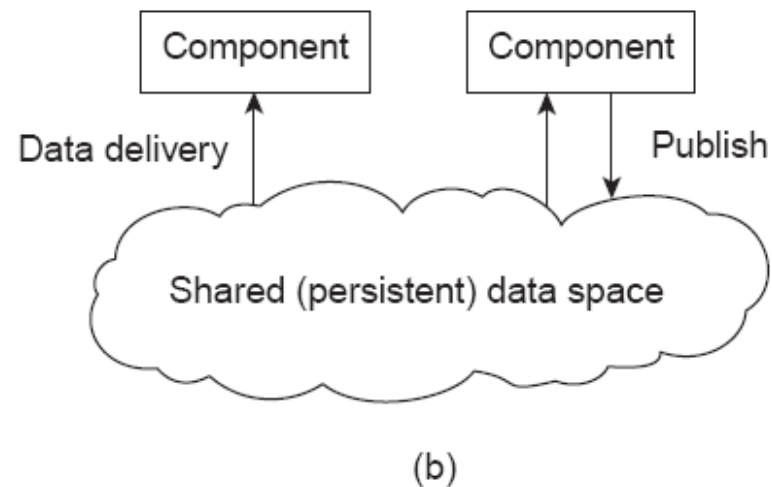


Sistema de Arquivos

- Os processos compartilham e trocam informações através do arquivos
 - Alguns grids computacionais, sistemas Web funcionam dessa maneira
 - É possível usar sistemas de arquivos distribuídos ou de rede
 - NFS, AFS
- Vantagens
 - Simplicidade
- Desvantagens
 - Segurança
 - Difícil pesquisa

Bases de Dados

- Processos se comunicam por meio de repositório comum (passivo ou ativo);
 - Normalmente, através de um SGBD
- Vantagens
 - Interface rica
 - Facilidade
- Desvantagens
 - Gerência



Formas Baseadas em Rede

- Os processos compartilham informações através de troca de mensagens (passing messages).
 - Neste modelo, o S.O é responsável pelo mecanismo de comunicação entre os processos.
- Necessário escolher o modelo de comunicação e o protocolo a ser utilizado.
 - Por que????

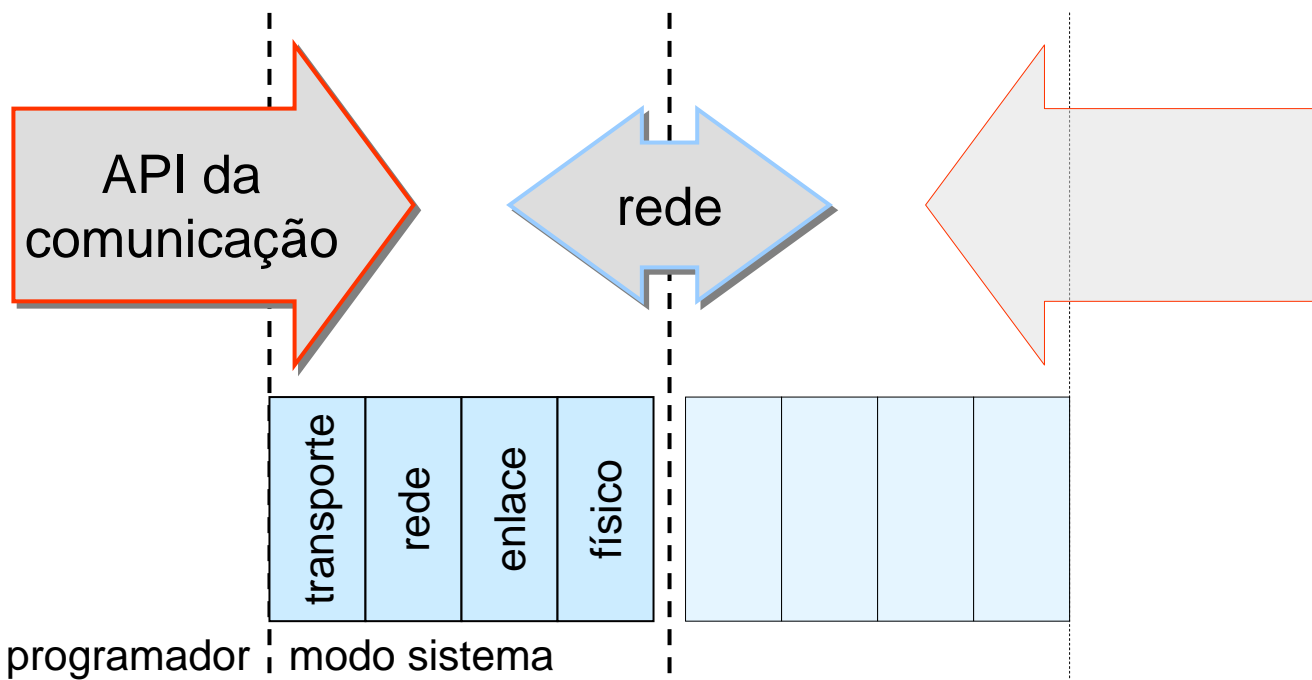
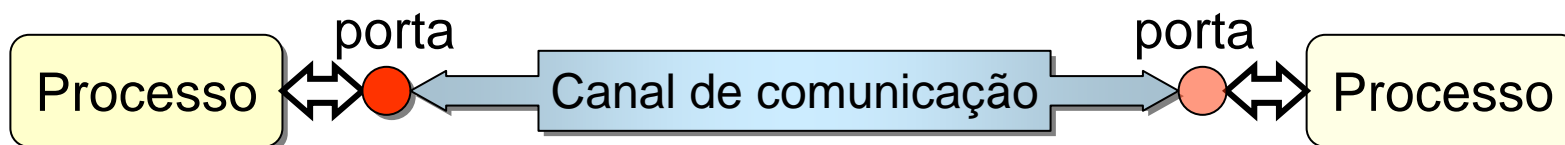
Modelos de Comunicação

- Modelos de comunicação entre aplicações
 - Difusão (“*broadcast*”) : o emissor envia a mesma informação a todos os restantes processos receptores.
 - Produtor-consumidor: comunicação uni-direccional, com o produtor a entregar mensagens ao consumidor.
 - Cliente-servidor : cliente controla totalmente o servidor.
 - Caixa de correio : as mensagens podem ser lidas por um processo receptor sem o emissor (um entre vários) poder controlar o processo que recolhe a mensagem.
 - Diálogo : dois processos acordam canal temporário para troca de mensagens durante uma sessão

Protocolos de Comunicação

- **Protocolo : regras de troca de mensagens entre os intervenientes.**
 - As partes envolvidas precisam concordar em utilizar o mesmo idioma!!!
- Exemplos de protocolos definidos no modelo cliente-servidor:
 - HTTP para WWW.
 - SMTP para transferência de Email entre nós
 - POP3 para utilizador recolher mensagens Email guardadas num ISP-Internet Service Provider.

Programação da comunicação: modelo



Primitivas Básicas

- **send:**
 - utilizada por um processo para enviar uma msg. Pode ser bloqueante ou não bloqueante.
- **receive:**
 - utilizada por um processo para receber uma msg. Essa operação é bloqueante, o processo que a executa fica bloqueado até o recebimento de uma mensagem.
- **reply:**
 - usada por um processo para responder a uma mensagem recebida. Pode conter uma informação de resposta ou simplesmente pode indicar que uma mensagem enviada com uma operação send foi recebida sem erros. A operação reply é não bloqueante.



IMAGE BANK
IMAGE BANK

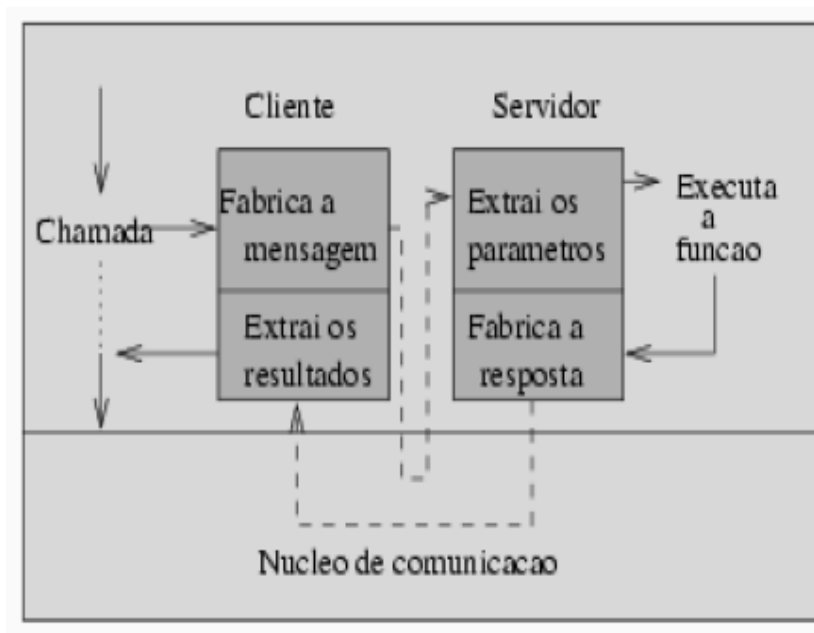
**Isso quer dizer que vocês
vão ter que enviar as
mensagens na mão?**

Chamadas de Procedimento Remoto (RPC)

- Permite a chamada de um procedimento em uma máquina por um programa sendo executado em outra máquina.
- Funcionamento:
 1. Um programa em uma máquina P, (o cliente) chama um procedimento em uma máquina Q, (o servidor) enviando os parâmetros adequados.
 2. O cliente é suspenso e a execução do procedimento começa.
 3. Os resultados são enviados da máquina Q para a máquina P e o cliente é acordado.

Chamadas de Procedimento Remoto (RPC)

- Facilita a vida do programador que não tem que se preocupar em *marshalling* dos dados



– O que é mesmo *marshalling*?

Chamada de Procedimento Remoto

- Principais problemas:
 - Chamador e procedimento executam em máquinas diferentes, executando em espaços de endereço diferentes;
 - Necessário passar parâmetros e resultados;
 - Em caso de falhas, cada uma das possíveis falhas causa problemas diferentes.
 - Problemas com firewall

Comunicação Orientada a Mensagem

- Quando não se pode adotar que o lado receptor está executando no momento em que uma requisição é emitida, o que fazer?
 - Troca de mensagens.
- Interface Berkeley
 - Interface Sockets
 - XTI (X/Open Transport Interface)
- Interface de troca de mensagens (MPI)

Interface de troca de mensagens (MPI)

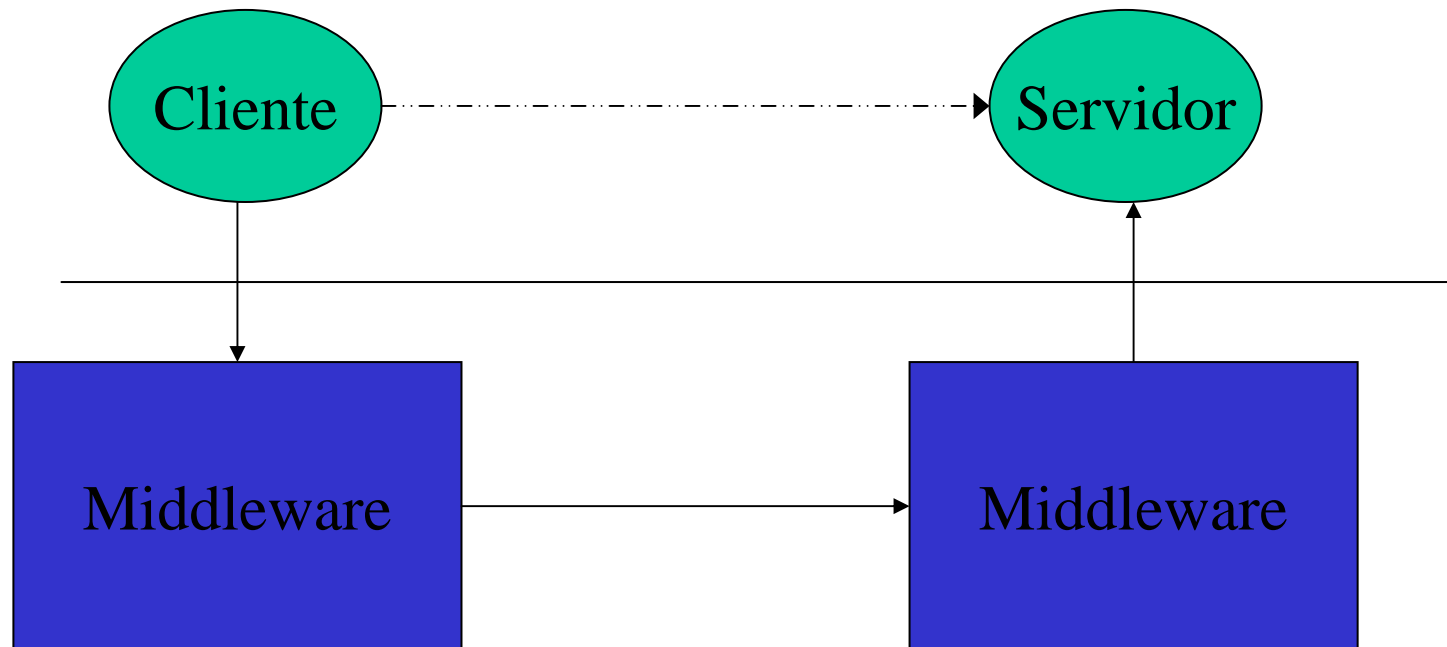
- MPI foi projetada para aplicações paralelas, para comunicação transiente, fazendo uso direto da rede subjacente, e considera que falhas sérias como quedas de processos ou partições da rede sejam fatais e não requeiram recuperação automática.
- A comunicação ocorre dentro de um grupo conhecido de processos.

Middleware

- Conjunto reusável e expansível de serviços e funções comumente necessários por parte de várias aplicações para funcionarem bem em um ambiente de rede.
- Modelos
 - Message-Oriented Middleware (MOM)
 - MPI
 - Object-Oriented Middleware
 - RMI

Middleware

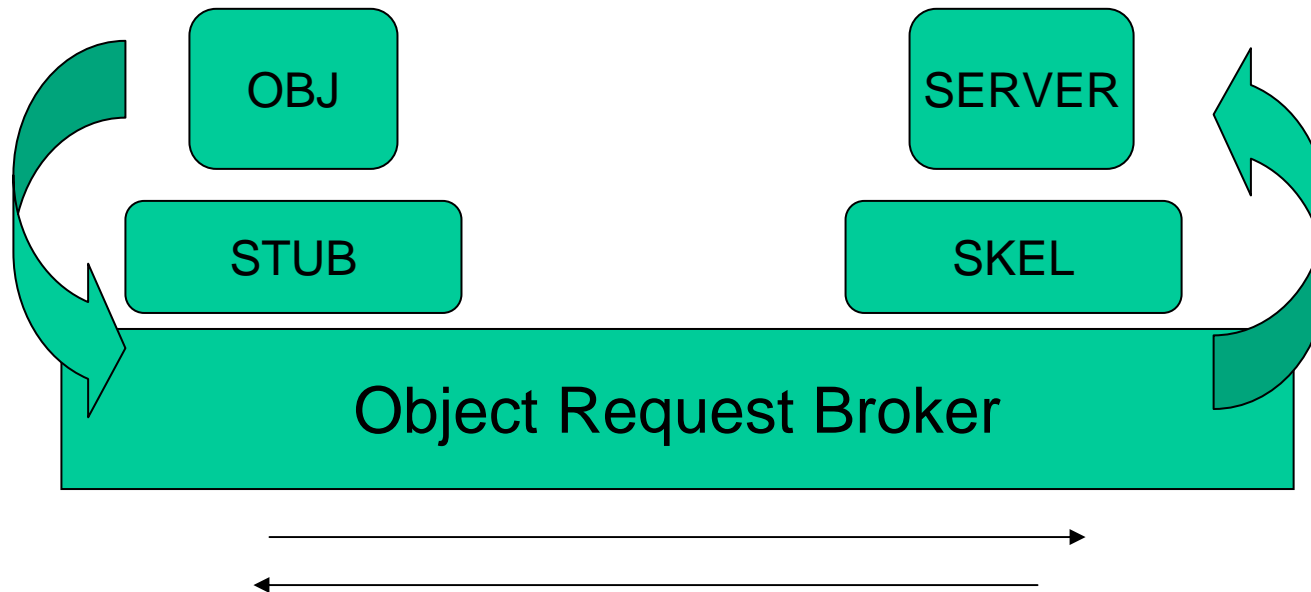
- Facilita o desenvolvimento de aplicações distribuídas.



Object-Oriented Middleware

- Permite objetos invocar transparentemente métodos de outros objetos (que podem estar em máquinas diferentes – objetos remotos)
- Libera o programador de tratar de detalhes como endereçamento e codificação/decodificação de mensagens

Object-Oriented Middleware



Middleware Orientado a Mensagens

- Comunicação distribuída baseada em um modelo de interação assíncrona
- A comunicação é efectuada através de mensagens que são transmitidas, ou pela estrutura cliente/servidor pura (usando broadcast ou multicast), ou entre pilhas
- Vantagens:
 - Comunicação assíncrona. A MOM usa uma aproximação “send and forget”.
- Desvantagens:
 - Modelo de programação não trivial

Middleware Orientado a Mensagens

- Se comunicam inserindo mensagens em filas específicas:
- O remetente tem a garantia que, a certa altura, sua mensagem será inserida na fila do receptor.
 - Não há garantias **QUANDO** ou **SE** a mensagem será lida pelo receptor!
- Comunicação fracamente acoplada em relação ao tempo.

Outras Formas de Comunicação

- Fluxo de dados
 - A temporização desempenha papel crucial.
 - Fluxo de áudio (mono/stereo);
 - Fluxo de vídeo e áudio;
 - Fluxos de dados são, portanto, seqüências unidades de dados.
- Comunicação Multicast



IMAGE BANK
IMAGE BANK

Qual mecanismo escolher?

**Você precisa analisar o problema para
depois escolher!**



Questões de implementação

- Como são estabelecidos os canais?
- Um canal pode estar associado a mais de dois processos?
- Quantos canais podem existir entre cada par de processos?
- Qual é a capacidade de um canal?
- Um canal pode acomodar mensagens de tamanho variáveis?
- O canal é unidirecional ou bidirecional?