

---

---

## Sistemas Distribuídos

RPC – Remote Procedure Call  
(Chamada Remota de  
Procedimento)

---

---

## Pesquisa

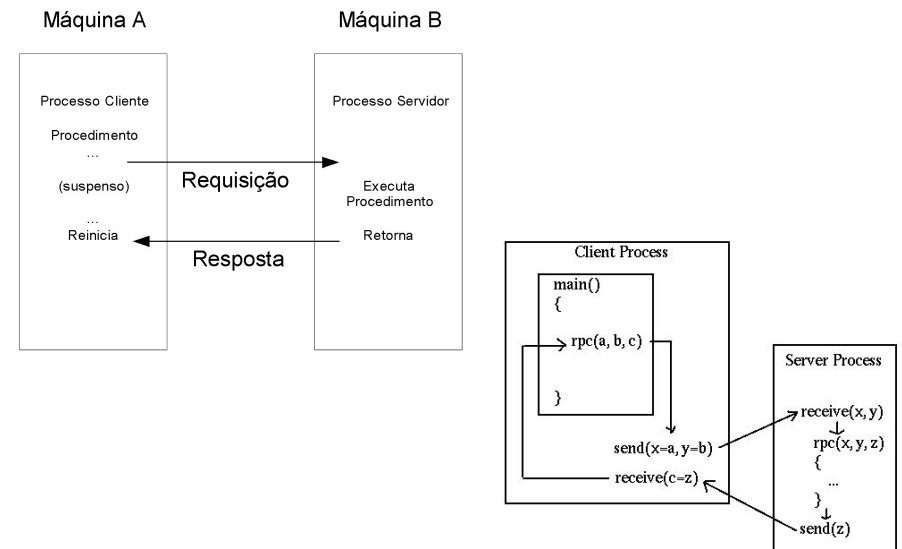
- Responda as seguintes perguntas:
  - O que é chamada de função (procedimento/método)
  - O que acontece no programa quando é executada uma chamada desse tipo?



## Conceito

- Chamada de procedimentos que estão em outras máquinas
  - Transferência de parâmetros para execução remota
  - Retorno das informações para o processo “chamador”
  - Conhecido como **chamada remota de procedimento** ou **RPC**
  - Transparência

## Conceito



## Problemas

- Espaços de endereçamento diferentes (máquinas diferentes)
- Arquitetura diferentes, interferindo nos tipos de dados
- Problemas no servidor
- Problemas no cliente

## Parâmetros

- Ponteiros
  - Cópia do endereço?
  - Utilização de técnicas cópia/restaura
    - Envio do buffer da mensagem, a partir do tamanho
  - Aumento do desempenho
    - Diminuição de um envio de mensagem, baseado no tipo do procedimento
- Não existe variáveis "globais"

## Características

- Funcionamento

- Para o SO (kernel), o envio e recebimento de mensagens é transparente, ou seja, ele não sabe que é uma RPC
- Portanto, deve existir um processo específico para o tratamento de RPCs

## Características

- Funcionamento

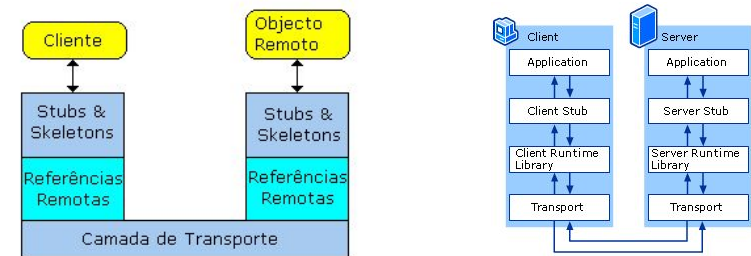
- Chamada ao *stub do cliente*
  - Versão no cliente responsável por iniciar a RPC
  - Esta versão recebe os parâmetros do procedimento e empacota para envio ao servidor, usando *send*
  - Após o envio, fica bloqueado (em *receive*) esperando a resposta do servidor

## Características

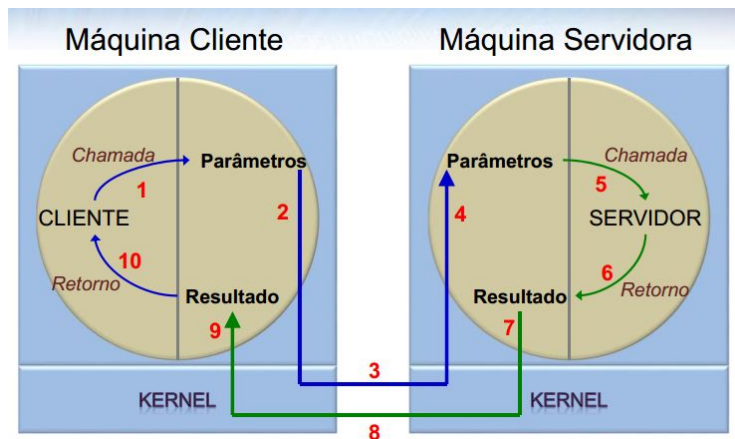
### ● Funcionamento

- No servidor, existe o *stub do servidor*
- Serviço de SO responsável por receber um pacote de informações remotas
- Verifica para qual serviço, e executa de acordo com os parâmetros recebidos
- Envia de volta ao cliente, através de *send*

## Funcionamento



## Características



## Características

- 1. procedimento chama o *stub de cliente* (forma normal)
- 2. stub constrói a mensagem e faz um *send* (trap ao kernel)
- 3. o kernel envia a mensagem para a máquina remota
- 4. o kernel remoto entrega a mensagem ao *stub do servidor*
- 5. o stub desempacota a mensagem e chama o "servidor"
- 6. o servidor realiza a tarefa e retorna ao stub
- 7. o stub do servidor empacota o retorno e faz um *send* para o cliente
- 8. o kernel remoto envia a mensagem ao kernel do cliente
- 9. o kernel do cliente entrega a mensagem ao stub do cliente
- 10. o stub desempacota a mensagem e entrega ao cliente

## Binding Dinâmico

- Ligação dinâmica entre Cliente/Servidor
- Especificação formal de um Servidor
  - IN: valores de ENTRADA no servidor
  - OUT: valores de SAÍDA do servidor
  - IN OUT: entrada e saída

## Binding

