

RPC - *Remote Procedure Call*

Sistemas Distribuídos e Mobile

Prof. Me. Gustavo Torres Custódio

gustavo.custodio@anhembi.br

Conteúdo

Remote Procedure Call (RPC)

RMI - Java

Exercícios

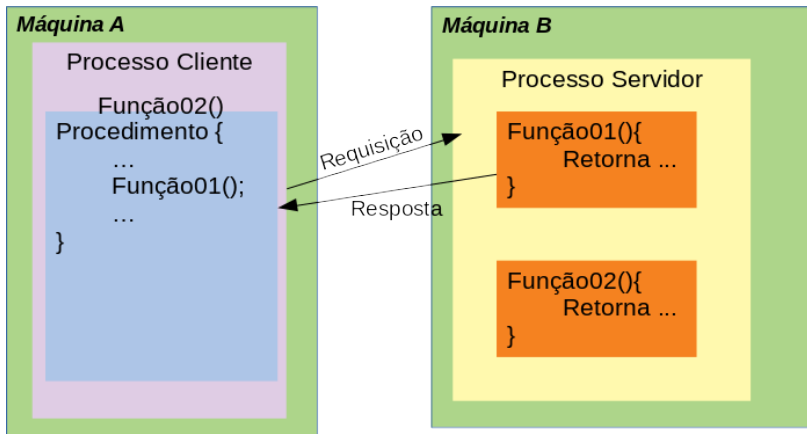


Remote Procedure Call (RPC)

RPC ~ Remote Procedure Call

- *Remote Procedure Call* (RPC) consiste em chamar um procedimento disponível em outra máquina.
 - Os parâmetros necessários para executar o procedimento são passados junto com a chamada remota.
 - Retorno das informações para o processo “chamador”.
 - Transparência - do ponto de vista do usuário, é como se um método normal fosse chamado.

RPC



Chamada de Procedimento Remoto

- Para o SO (kernel), o envio e recebimento de mensagens é transparente, ou seja, ele não sabe que está fazendo uma RPC.
 - Transparência.
 - Portanto, deve existir um processo específico para o tratamento de RPCs.
 - Isso é feito por meio de *stubs*.
 - Basicamente possuem a função de converter chamadas remotas para locais.

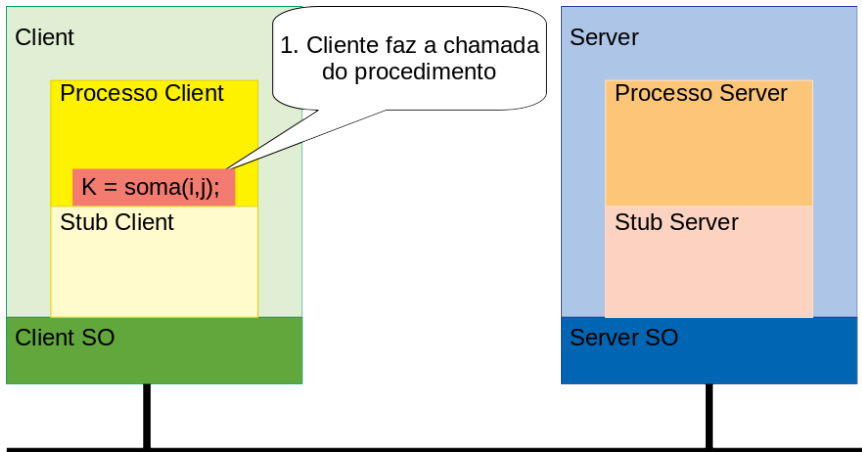
Chamada de Procedimento Remoto

- *Stub do Cliente:*
 - Empacota os parâmetros em uma mensagem e a envia para a máquina do servidor.
 - O cliente bloqueia a si mesmo em *receive* até que a resposta do servidor chegue (síncrono).

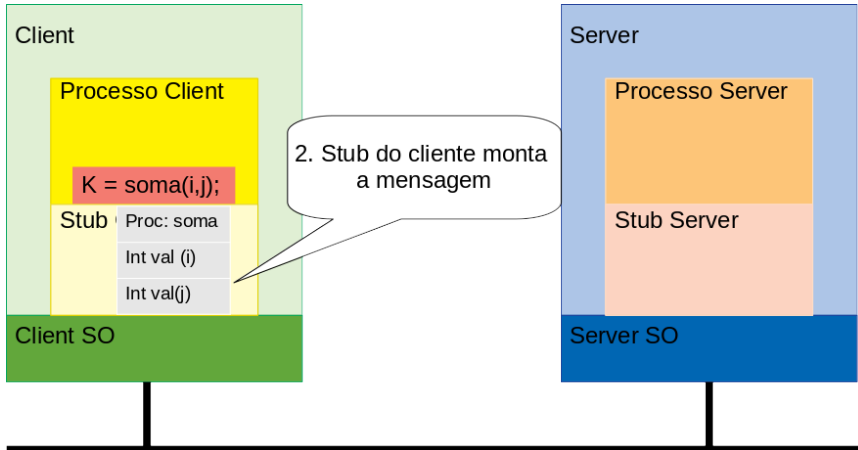
Chamada de Procedimento Remoto

- *Stub do Servidor:*
 - Quando a mensagem chega ao servidor, o SO dele passa a mensagem para o *stub* do servidor.
 - Desempacota os parâmetros da mensagem do cliente e executa o procedimento no servidor.
 - Envia mensagem de volta ao cliente por meio da instrução *send*.

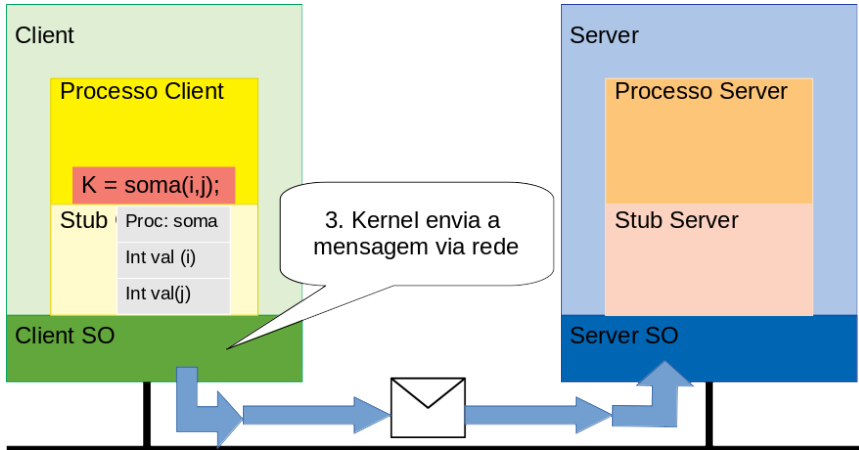
Chamada de Procedimento Remoto



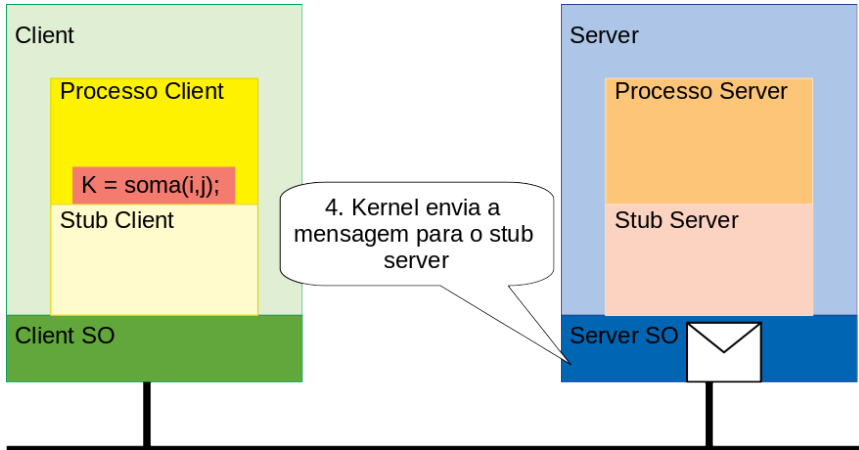
Chamada de Procedimento Remoto



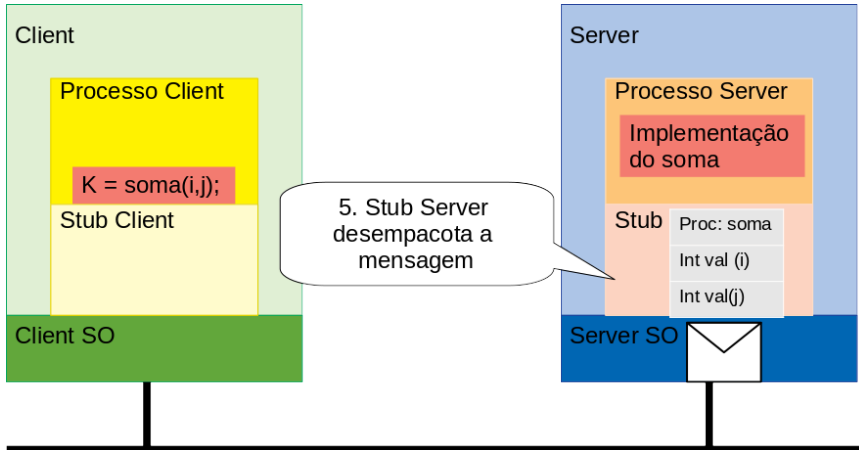
Chamada de Procedimento Remoto



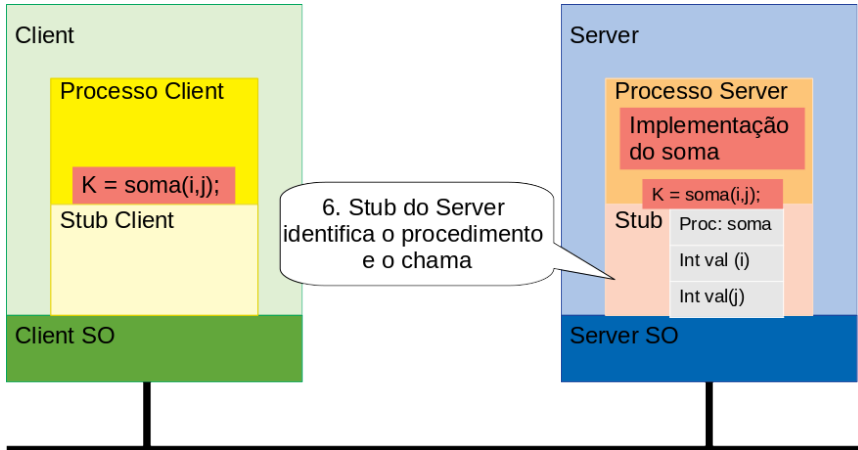
Chamada de Procedimento Remoto



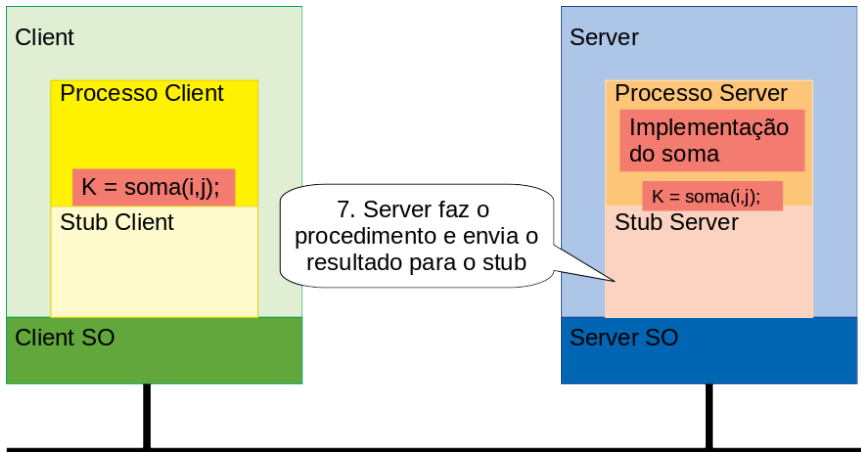
Chamada de Procedimento Remoto



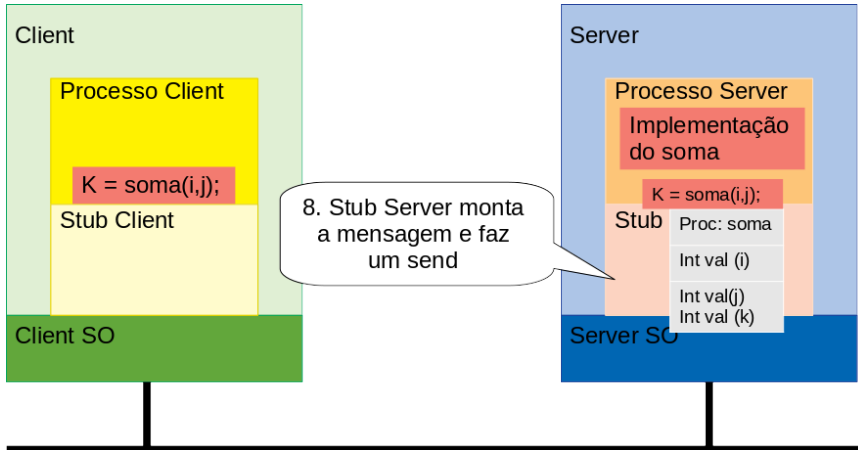
Chamada de Procedimento Remoto



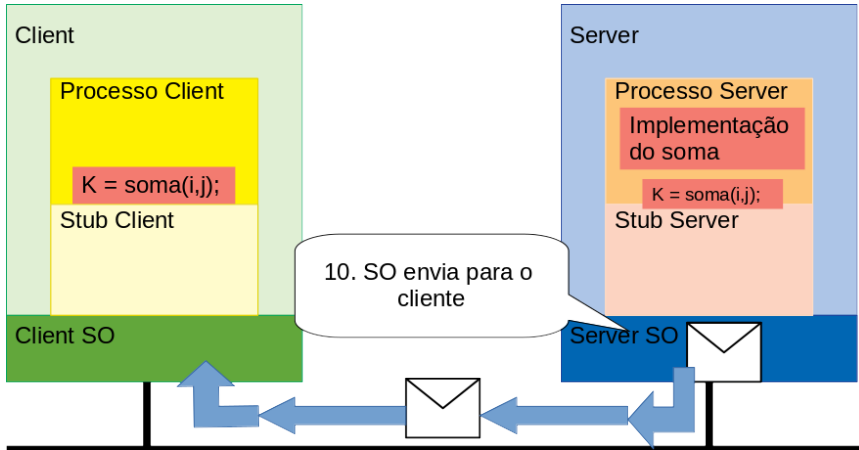
Chamada de Procedimento Remoto



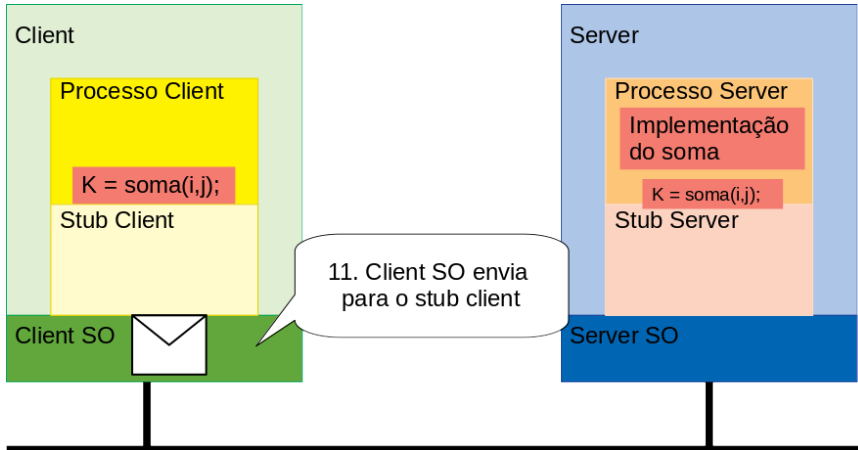
Chamada de Procedimento Remoto



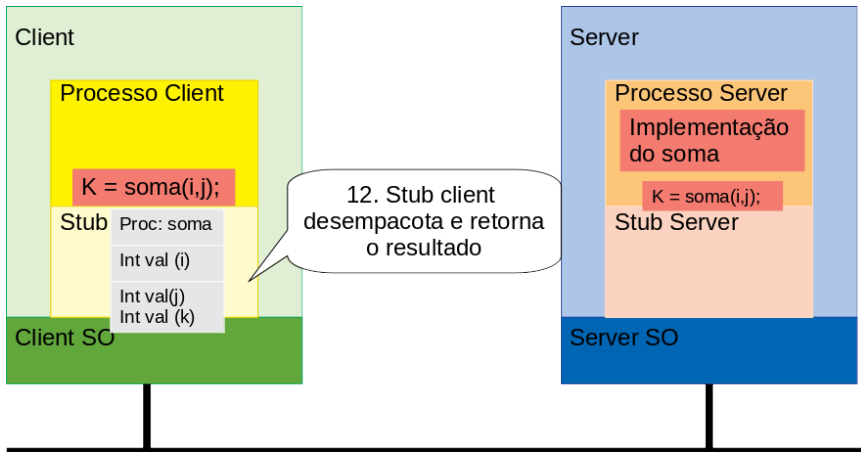
Chamada de Procedimento Remoto



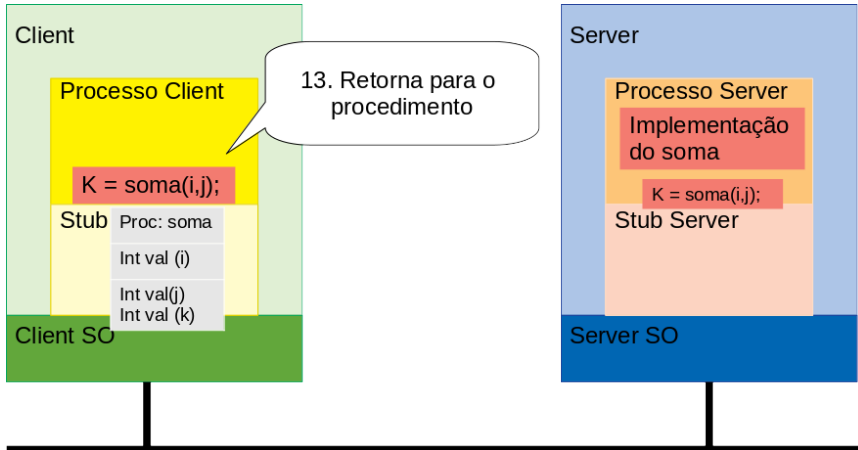
Chamada de Procedimento Remoto



Chamada de Procedimento Remoto



Chamada de Procedimento Remoto



Binding Dinâmico - RPC

- O *binder* funciona como um *nameserver* para o RPC.
- É necessário o cliente enviar uma mensagem para o *binder* para localizar o servidor
- É possível o uso de múltiplos *binders*.

Binding Dinâmico - RPC

- Servidor
 - Ao ser executado, avisa ao programa de registro (*binder*) que está ativo: registro do serviço.
- Cliente
 - Quando um procedimento é chamado, este se liga ao servidor, utilizando o *binder* como intermediário.



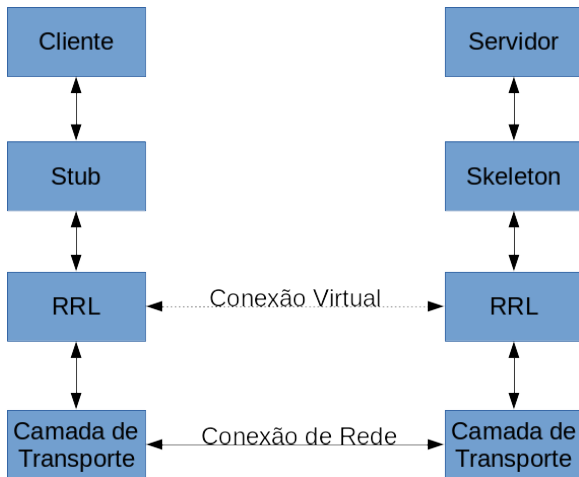
RPC ~ Remote Procedure Call

RMI - Java

RMI - Java

- Remote Method Invocation
 - RPC da linguagem Java;
 - Orientação a Objetos;
 - 3 camadas:
 - *Stub/Skeleton*: é invocado primeiro, recebendo objetos e serializando-os (bytes).
 - RRL – *Remote Reference Layer*: cada lado possui a sua, responsável por montar a mensagem de envio/resposta entre cliente/servidor.
 - Transporte: envio das informações via rede – TCP/IP.

RMI - Java



Tutorial Java - RMI

- Servidor
 - Interface que contém os métodos a serem disponibilizados para os clientes.
- implementa interface Remote
 - Classe que contém a lógica dos métodos, além das rotinas de registro.
- Herda características de UnicastRemoteObject
- Implementa a interface definida
- Método `main()` para registrar o serviço



RPC - Remote Procedure Call

Exercícios

Exercícios

- Baixe o arquivo:
 - `codigos_rpc.zip`

Exercício 1

- Altere o exemplo da calculadora para fazer as operações de soma, subtração, multiplicação e divisão, de acordo com a escolha do usuário.

Exercício 2

- Escreva uma aplicação dicionário.
 - O servidor conterà um arquivo com palavras em Português e Inglês separadas por vírgula.
 - Exemplo: sim,yes.
 - O Cliente irá chamar um método que recebe como parâmetro uma palavra em Português e retorna a tradução em Inglês.
 - Vamos fazer simplificações no dicionário, pois o conceito principal é a aplicação do RMI e não o dicionário propriamente dito.

Referências

- Livro:
 - Sistemas Distribuídos: Princípios e Paradigmas
 - Tanenbaum
 - Cap 10 - Sistemas distribuídos baseados em objetos
- Livro:
 - Sistemas Distribuídos: Conceitos e Projetos
 - Coulouris
 - Cap 5 - Invocação Remota

Conteúdo



<https://gustavotcustodio.github.io/sdmobile.html>

Obrigado

gustavo.custodio@anhembi.br