

Sistemas Distribuídos

Professor: Rodrigo da Rosa Righi

Contato: rrrighi@unisinos.br

Aula: 3

Dia/Horário: Sexta-Feira, 19:30 - 22:23

Agenda

- * Empacotamento e Representação de Dados
- * Serialização em Java
- * Tratamento de Arquivos XML
- * Chamada Remota de Procedimentos
- * Invocação Remota de Métodos
- * Programando com Sockets



Você se lembra da Aula 2?

Comunicação síncrona e
assíncrona

Comunicação One-sided

Sockets TCP

Sockets UDP

Pensando em fazer cursos Online?

<http://www.udacity.com>

Udacity

Udacity is a totally new kind of learning experience. You learn by solving challenging problems and pursuing udacious projects with world-renowned university instructors (not by watching long, boring lectures). At Udacity, we put you, the student, at the center of the universe. [Keep Reading](#)

The Four Elements of Udacity

- 1 Take any of our 14 classes. 100% free! [Course Catalog](#)
- 2 Join a community of 112,091 active students and instructors. 100% free!
- 3 Optionally certify your skills online or in one of our 4,500 testing centers, for a fee.
- 4 Optionally let us hand your resume to one of our 20 partner companies. 100% free for students!

Shortcuts to all our classes. Try out any of them:

Intro to Computer Science Building a Search Engine Beginner	Intro to Physics Landmarks in Physics Beginner
Intro to Statistics Making Decisions based on Data	Web Application Engineering How to Build a Blog Intermediate

Check out what it's like to take a course: Watch a course now.

Sign up to receive our monthly newsletter.

Udacity Blog

One more week left of the Secondary School Challenge; 10,000 units already completed
Mon, 20 Aug 2012

3 ways to make your Global Meetup Day event amazing!
Wed, 15 Aug 2012

3 new classes open for enrollment
Tue, 14 Aug 2012

Download your organizer packet for Udacity's Global Meetup Day
Sat, 11 Aug 2012

Finished your final? Get an official Udacity certificate
Wed, 08 Aug 2012

[Read more blog posts](#)

coursera

COURSES JOBS ABOUT ▾ LOGIN

Take the World's Best Courses, Online, For Free.

Join 1,128,109 Courserians

Absolutely loving that my Coursera classmates and I are chatting about our lives and experiences from all around the world! - Dawn Smith

Learn with videos, quizzes, and assignments

Interact with thousands of other students

Advance your knowledge and career

119 COURSES

[View All 119 Courses](#)

Cryptography Stanford University Starts in 4 days (6 weeks long)	Model Thinking University of Michigan Starts in 9 days (10 weeks long)	Securing Digital Democracy University of Michigan Starts in 11 days (5 weeks long)	Gamification University of Pennsylvania Starts in 4 days (6 weeks long)
--	--	--	---

ACROSS 16 CATEGORIES

[View All 119 Courses](#)

Biology & Life Sciences	Business & Management	Computer Science: Artificial Intelligence, Robotics, Vision
Computer Science: Programming & Software Engineering	Computer Science: Systems, Security, Networking	Computer Science: Theory
Economics & Finance	Education	Electrical and Materials Engineering
Health and Society & Medical Ethics	Humanities and Social Sciences	Information, Technology, and Design
Mathematics	Medicine	Physical & Earth Sciences
Statistics, Data Analysis, and Scientific Computing		

FROM 16 UNIVERSITIES

Penn	UNIVERSITY OF TORONTO	UNIVERSITY OF WASHINGTON	UCSF
STANFORD UNIVERSITY	THE UNIVERSITY OF EDINBURGH	RICE	JOHNS HOPKINS UNIVERSITY
Duke	Caltech	PRINCETON UNIVERSITY	ILLINOIS
EPFL	UNIVERSITY OF VIRGINIA	MICHIGAN STATE UNIVERSITY	Georgia Institute of Technology

Empacotamento e Representação de Dados

Estruturas de Dados

Objetos Interconectados

O que deve
ocorrer antes da
transmissão?

Sequência de
bytes

Serialização

Representação dos Dados

Organização na
memória

Big
Endian

Little
Endian

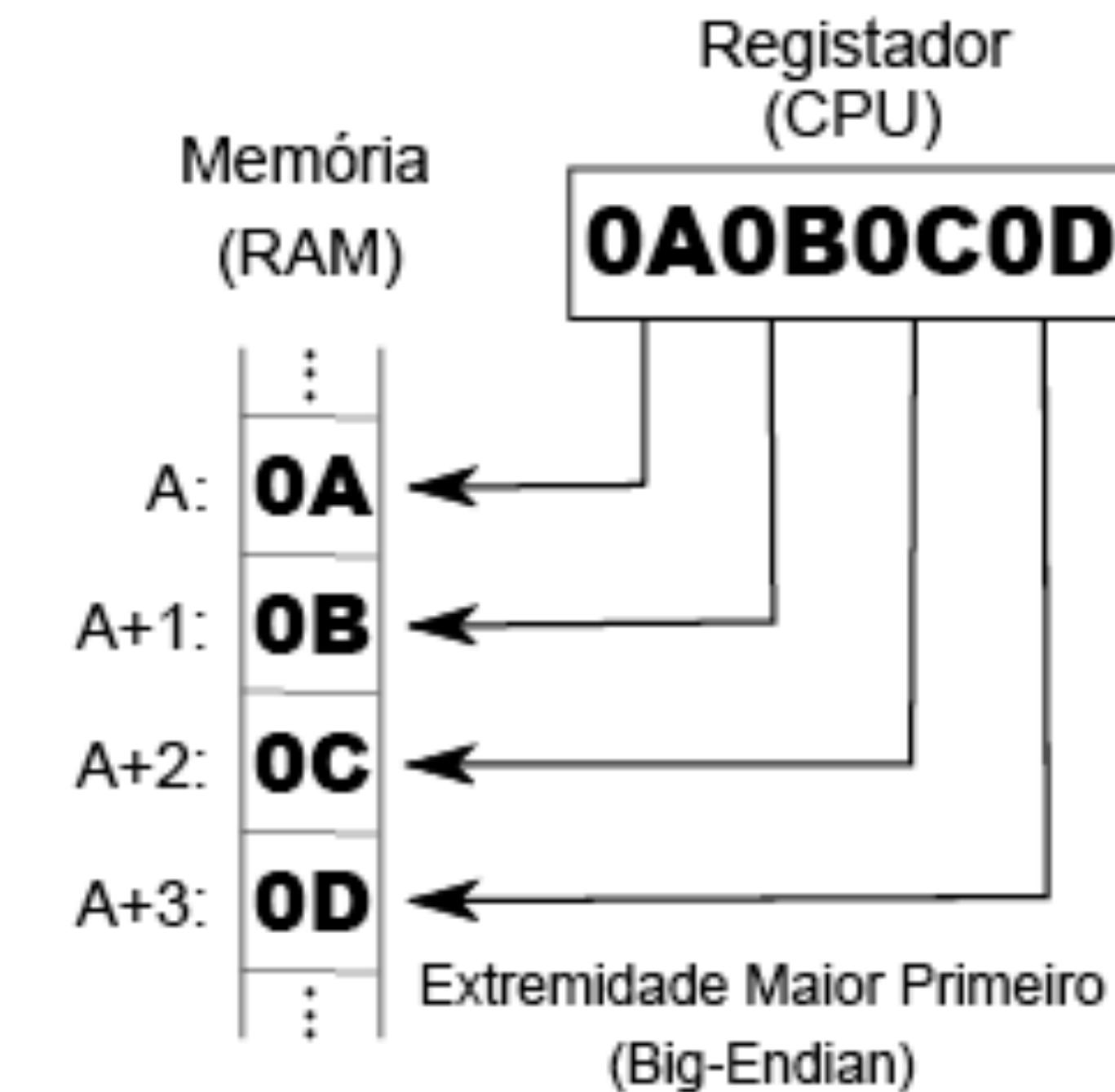
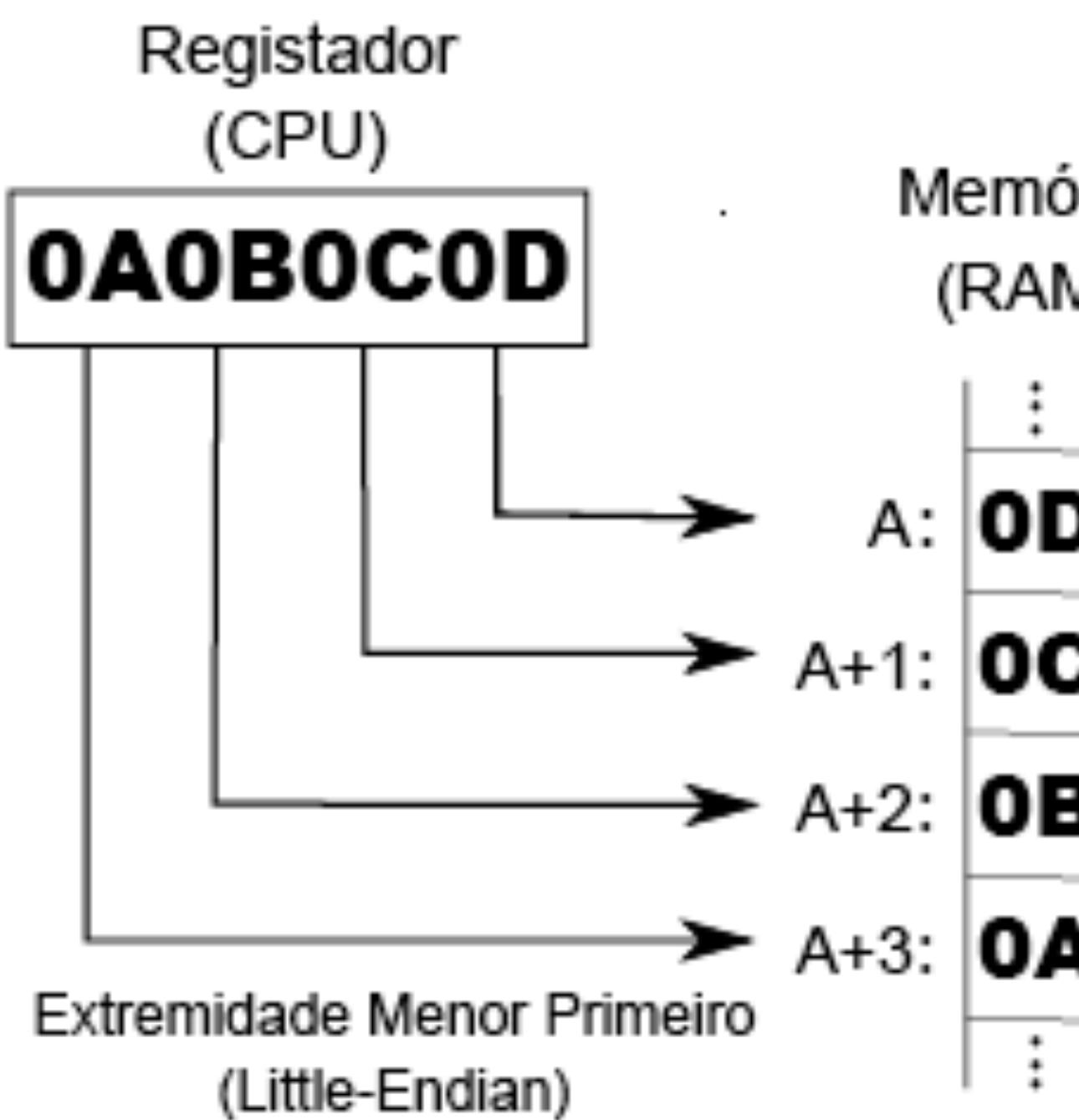
Caracteres

ASCII

Unicode

Empacotamento e Representação de Dados

Organização na Memória



Empacotamento e Representação de Dados

Tabela ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	NUL	(null)	32	20 040	 	Space		64	40 100	@	Ø	96	60 140	`	`	~	
1	1 001	SOH	(start of heading)	33	21 041	!	!	!	65	41 101	A	A	97	61 141	a	a	ä	
2	2 002	STX	(start of text)	34	22 042	"	"	"	66	42 102	B	B	98	62 142	b	b	ö	
3	3 003	ETX	(end of text)	35	23 043	#	#	#	67	43 103	C	C	99	63 143	c	c	ç	
4	4 004	EOT	(end of transmission)	36	24 044	$	\$	\$	68	44 104	D	D	100	64 144	d	d	đ	
5	5 005	ENQ	(enquiry)	37	25 045	%	%	%	69	45 105	E	E	101	65 145	e	e	é	
6	6 006	ACK	(acknowledge)	38	26 046	&	&	&	70	46 106	F	F	102	66 146	f	f	đ	
7	7 007	BEL	(bell)	39	27 047	'	'	'	71	47 107	G	G	103	67 147	g	g	ğ	
8	8 010	BS	(backspace)	40	28 050	(((72	48 110	H	H	104	68 150	h	h	ı	
9	9 011	TAB	(horizontal tab)	41	29 051)))	73	49 111	I	I	105	69 151	i	i	ı	
10	A 012	LF	(NL line feed, new line)	42	2A 052	*	*	*	74	4A 112	J	J	106	6A 152	j	j	ı	
11	B 013	VT	(vertical tab)	43	2B 053	+	+	+	75	4B 113	K	K	107	6B 153	k	k	ı	
12	C 014	FF	(NP form feed, new page)	44	2C 054	,	,	,	76	4C 114	L	L	108	6C 154	l	l	ı	
13	D 015	CR	(carriage return)	45	2D 055	-	-	-	77	4D 115	M	M	109	6D 155	m	m	ı	
14	E 016	SO	(shift out)	46	2E 056	.	.	.	78	4E 116	N	N	110	6E 156	n	n	ı	
15	F 017	SI	(shift in)	47	2F 057	/	/	/	79	4F 117	O	O	111	6F 157	o	o	ı	
16	10 020	DLE	(data link escape)	48	30 060	0	Ø	Ø	80	50 120	P	P	112	70 160	p	p	ı	
17	11 021	DC1	(device control 1)	49	31 061	1	1	1	81	51 121	Q	Q	113	71 161	q	q	ı	
18	12 022	DC2	(device control 2)	50	32 062	2	2	2	82	52 122	R	R	114	72 162	r	r	ı	
19	13 023	DC3	(device control 3)	51	33 063	3	3	3	83	53 123	S	S	115	73 163	s	s	ı	
20	14 024	DC4	(device control 4)	52	34 064	4	4	4	84	54 124	T	T	116	74 164	t	t	ı	
21	15 025	NAK	(negative acknowledge)	53	35 065	5	5	5	85	55 125	U	U	117	75 165	u	u	ı	
22	16 026	SYN	(synchronous idle)	54	36 066	6	6	6	86	56 126	V	V	118	76 166	v	v	ı	
23	17 027	ETB	(end of trans. block)	55	37 067	7	7	7	87	57 127	W	W	119	77 167	w	w	ı	
24	18 030	CAN	(cancel)	56	38 070	8	8	8	88	58 130	X	X	120	78 170	x	x	ı	
25	19 031	EM	(end of medium)	57	39 071	9	9	9	89	59 131	Y	Y	121	79 171	y	y	ı	
26	1A 032	SUB	(substitute)	58	3A 072	:	:	:	90	5A 132	Z	Z	122	7A 172	z	z	ı	
27	1B 033	ESC	(escape)	59	3B 073	;	;	;	91	5B 133	[[123	7B 173	{	{	ı	
28	1C 034	FS	(file separator)	60	3C 074	<	<	<	92	5C 134	\	\	124	7C 174	|		ı	
29	1D 035	GS	(group separator)	61	3D 075	=	=	=	93	5D 135]]	125	7D 175	}	}	ı	
30	1E 036	RS	(record separator)	62	3E 076	>	>	>	94	5E 136	^	^	126	7E 176	~	~	ı	
31	1F 037	US	(unit separator)	63	3F 077	?	?	?	95	5F 137	_	_	127	7F 177		DEL	ı	

Empacotamento e Representação de Dados

Métodos para permitir que os computadores troquem informações binárias

1 - Valores são convertidos para um formato externo

Caso dois computadores sejam conhecidos, há otimização?

Formato conhecido antes da transmissão e convertido para a forma local na recepção

2 - Os valores são transmitidos de acordo com o formato do transmissor

Informa na mensagem o formato usado

Empacotamento e Representação de Dados

Empacotamento

Marchalling

Processo de pegar uma coleção de itens de dados e montá-los de uma forma escolhida para transmissão

Desempacotamento

Unmarshalling

Processo de desmonstagem na chegada para produzir a coleção de dados

Alternativas

CORBA

XML

Serialização
de objetos
em Java

Empacotamento e Representação de Dados

Java RMI

Ambos objetos e valores de dados primitivos pode ser passados como argumentos ou resultado de uma invocação de métodos

Objeto

Instância de uma classe

Para serem transmitidos pela rede, devem implementar a Interface

Serializable

Serialização em Java

Serialização

Passar um objeto ou um conjunto de objetos para uma forma serial que pode ser gravada em disco ou transmitida em uma mensagem.

O processo que faz a deserialização não possui conhecimento prévio dos tipos de objetos na forma serializada.

Assim, algumas informações sobre a classe do objeto devem ser incluídas na forma serializada. Esta informação habilita o receptor a carregar a classe apropriada quando um objeto é deserializado.

Reflexão

Serialização em Java

Reflexão

Indagar sobre propriedades de uma classe

Permite que classes sejam criadas a partir de seus nomes

Serialização usa reflexão de modo recursivo

Tratamento de Arquivos XML

XML

Linguagem de
marcações

W3C (World Wide
Web Consortium)

Estruturas
lógicas de dados

Extensível

Dados lidos
por humanos

XML e Web
Services

Comunicação entre cliente e servidor

Descrição de Web Services

XML

Web Services

SOAP

Normalmente, HTTP

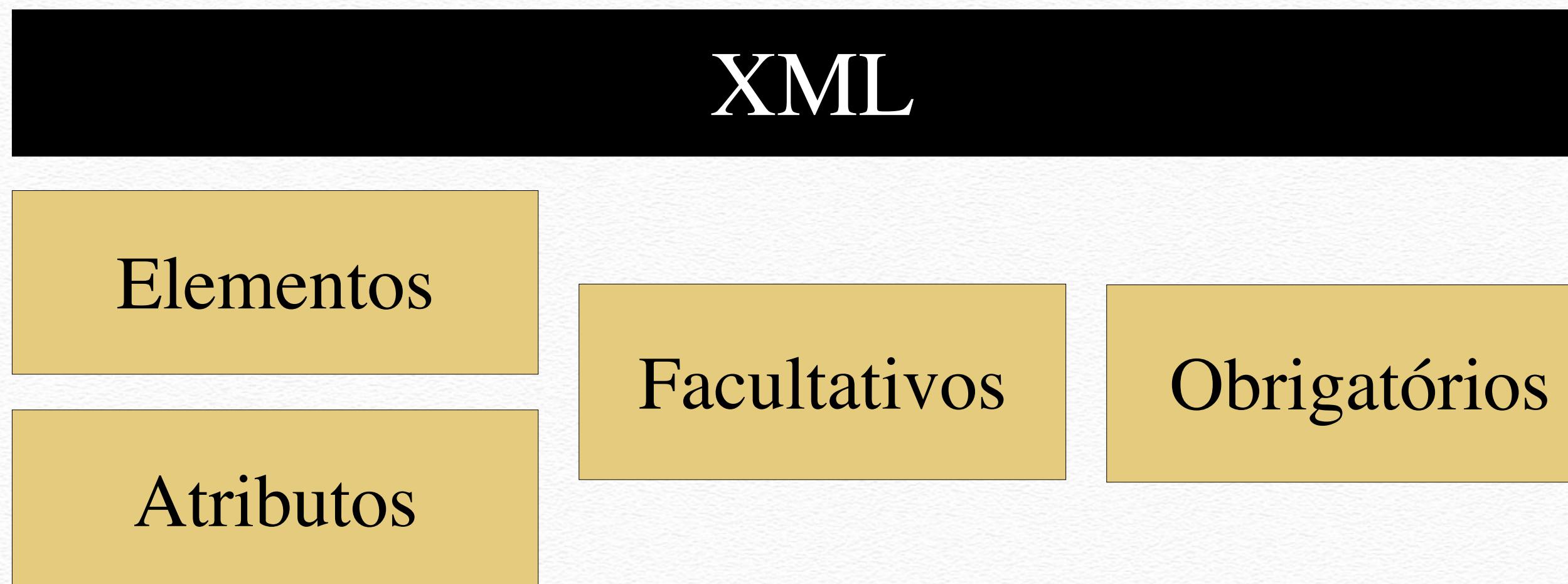
Tratamento de Arquivos XML

XML

Normalmente, as mensagens são muito grandes, fato que requer poder de processamento e tempo de transmissão maiores.

Além disso, necessita-se de mais espaço em memória para armazenar arquivos XML.

Tratamento de Arquivos XML



```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

XML
Schema

Tratamento de Arquivos XML

Dados binários em XML

Toda a informação deve ser transmitida através de caracteres

Qual a
alternativa?

Base 64

Usa caracteres alfanuméricos junto com + , / =
que possuem suginicado especial

Mensagens XML não são indicadas para a
computação de alto desempenho

Chamada Remota de Procedimentos

send()

Abstração de baixo nível

receive()

Desvantagem

Remote Procedure Call

1984

Permitir que programas procedurais chamassem procedimentos localizados em outras máquinas

Chamada Remota de Procedimentos

Palavra-chave

Transparência

Empacotamento e trocas de mensagens são ocultadas do programador

Componentes

Stub de cliente

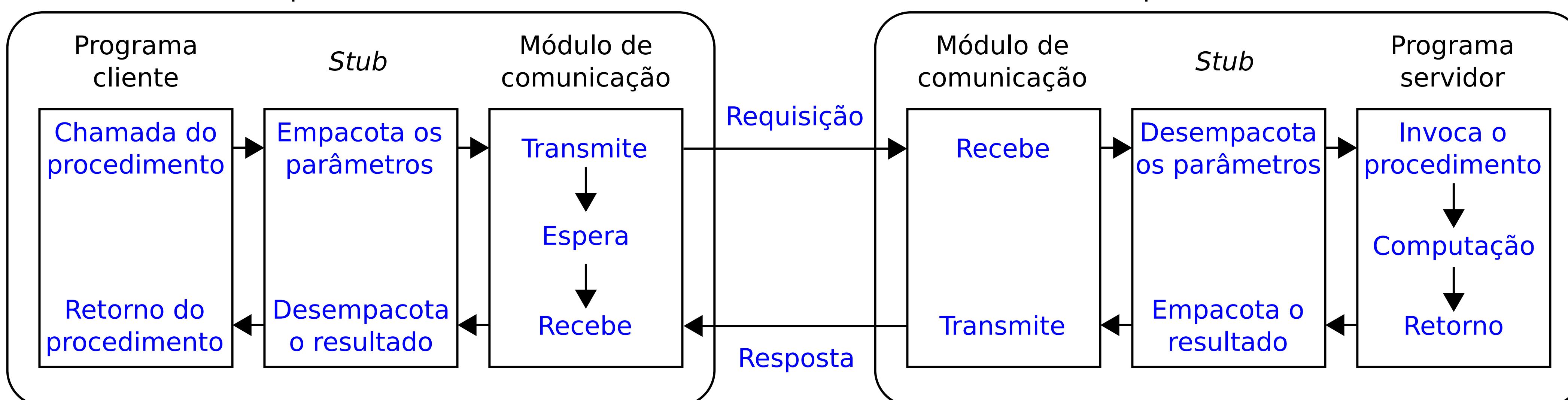
Módulo de Comunicação

Stub de servidor

Computador cliente

Rede

Computador servidor



Chamada Remota de Procedimentos

Módulo de
Comunicação

Cooperam entre si para executar o protocolo requisição-resposta

Stub de
Cliente

Ao invés de executar o procedimento, faz o empacotamento e envia os dados para o servidor por meio do módulo de comunicação

Stub de
servidor

Quando a mensagem de resposta chega, ele desempacota os resultados

Desempacotar os argumentos da mensagem de requisição, invocar o procedimento correspondente e empacotar os valores de retorno na mensagem de resposta

Chamada Remota de Procedimentos

RPC

Servidor fornece um conjunto
de procedimentos disponíveis

Servidor define uma
interface de serviço

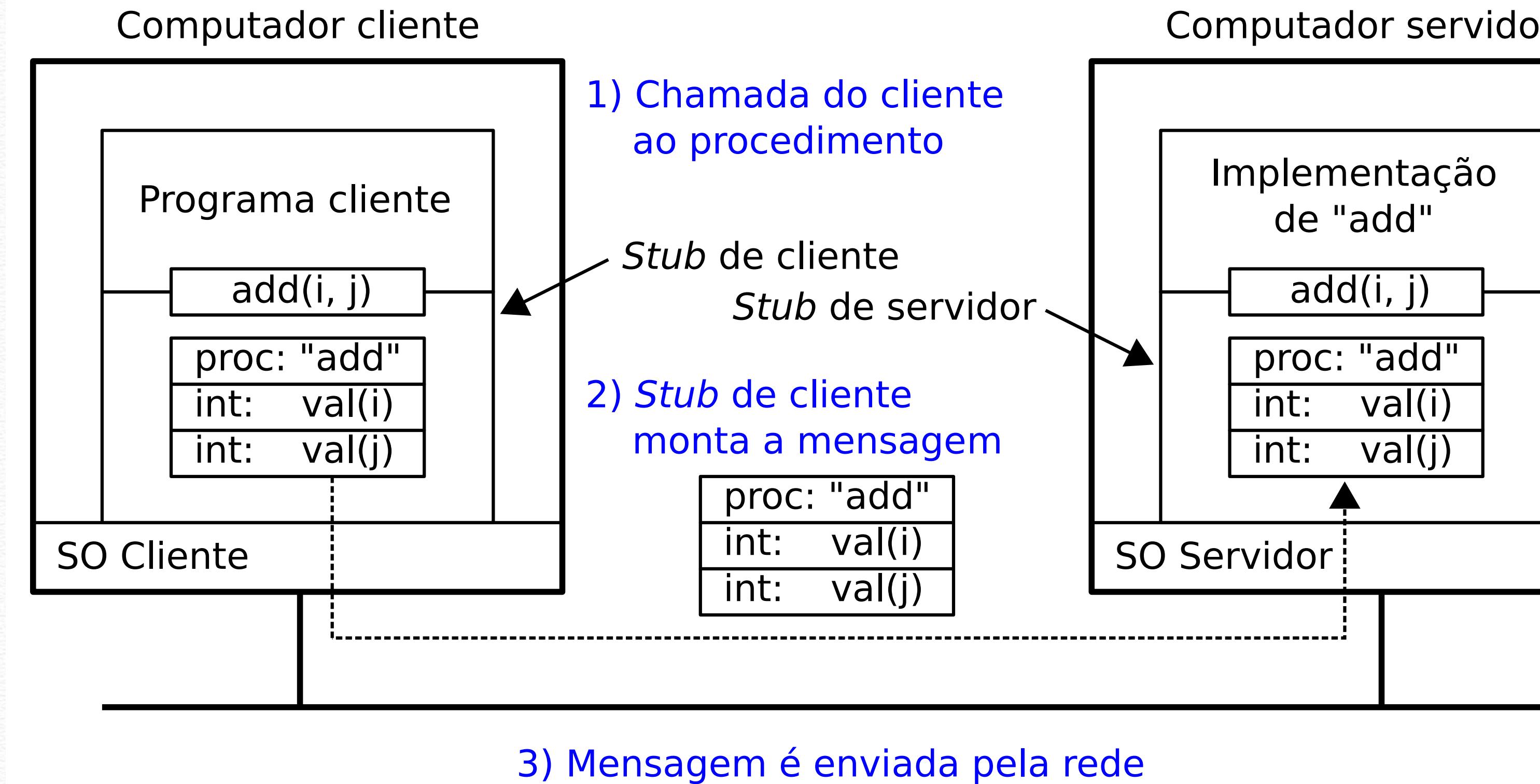
Interface define as assinaturas do
conjunto de procedimentos

Linguagem de
Definição de
Interface

IDL

Chamada Remota de Procedimentos

Passagem por valor



Passagem por valor

Passagem por referência

Chamada Remota de Procedimentos

Passagem por
referência

Como
resolver?

Cada computador tem a sua memória e seus dados

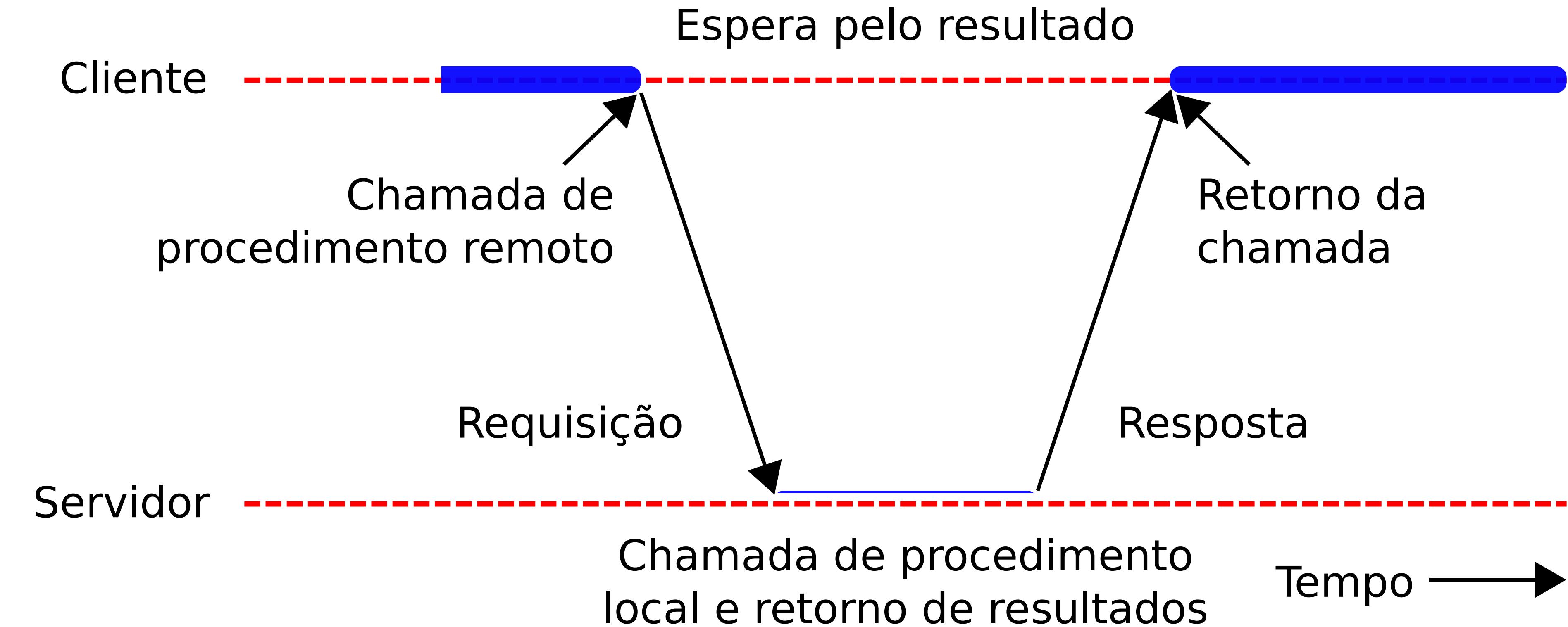
Ponteiros de memória não são válidos na máquinas remota

1 - Proibir passagem de ponteiros e parâmetros de referência

2 - Copiar o valor apontado pela variável de referência, fazer as alterações remotamente e depois atualizar a variável no chamador

Chamada Remota de Procedimentos

Comunicação Síncrona

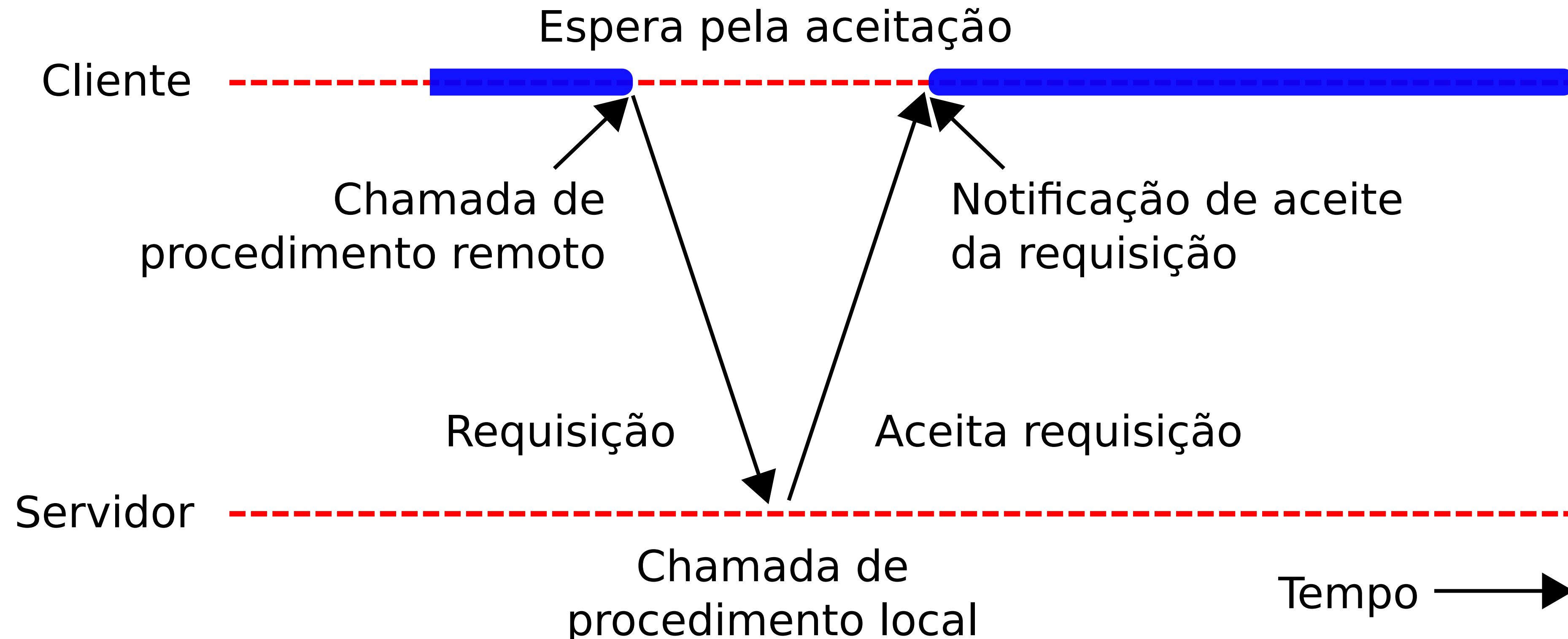


Chamada Remota de Procedimentos

Comunicação Assíncrona

Tipo 1

Servidor envia resposta dizendo que aceitou a requisição



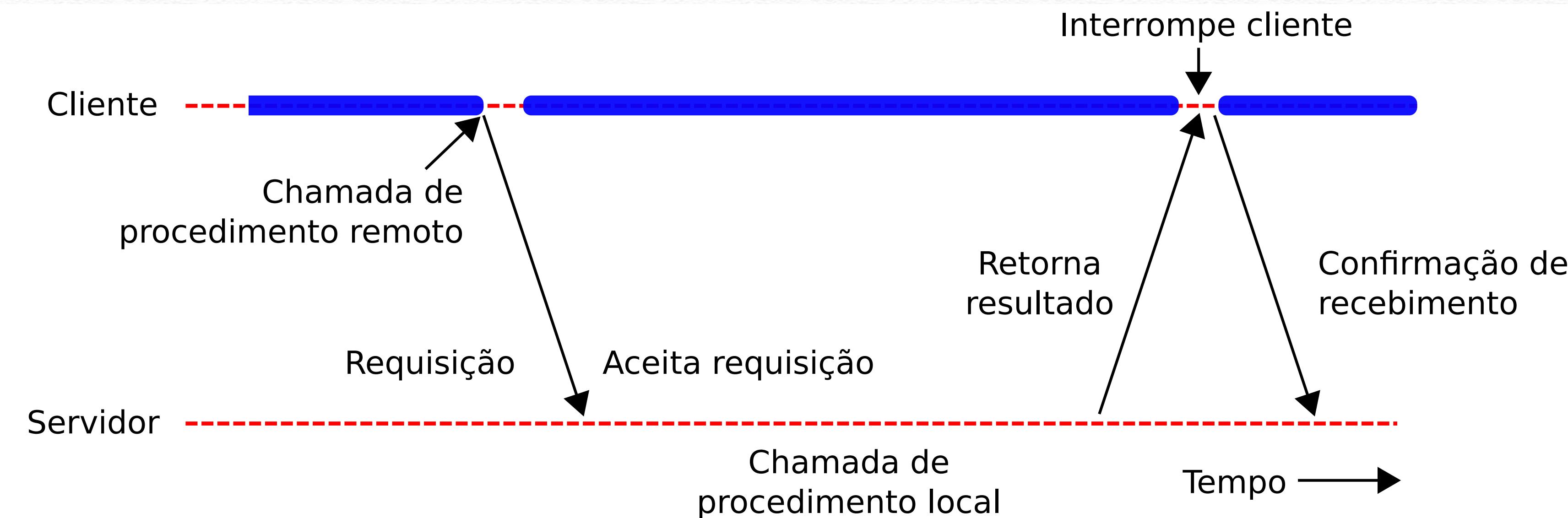
Chamada Remota de Procedimentos

Comunicação Assíncrona

Tipo 2

Cliente não espera pelo
reconhecimento

RPC de 1 via



Problema?

Quando a confiança não é garantida, o cliente pode não saber com certeza se a requisição será ou não passada

Invocação Remota de Métodos

RMI (Remote Method Invocation)

Analogia com RPC

Arquitetura Cliente-servidor

Idéia-chave

Transparência

Invocação Remota de Métodos

Componentes básicos

Proxy

Análogo ao Stub
de RPC

Torna transparente a
invocação a métodos
remotos

Oculta detalhes

Referência ao
objeto remoto

Empacotamen-
to e desempa-
cotamento de
dados

Skeleton

Análogo ao Stub

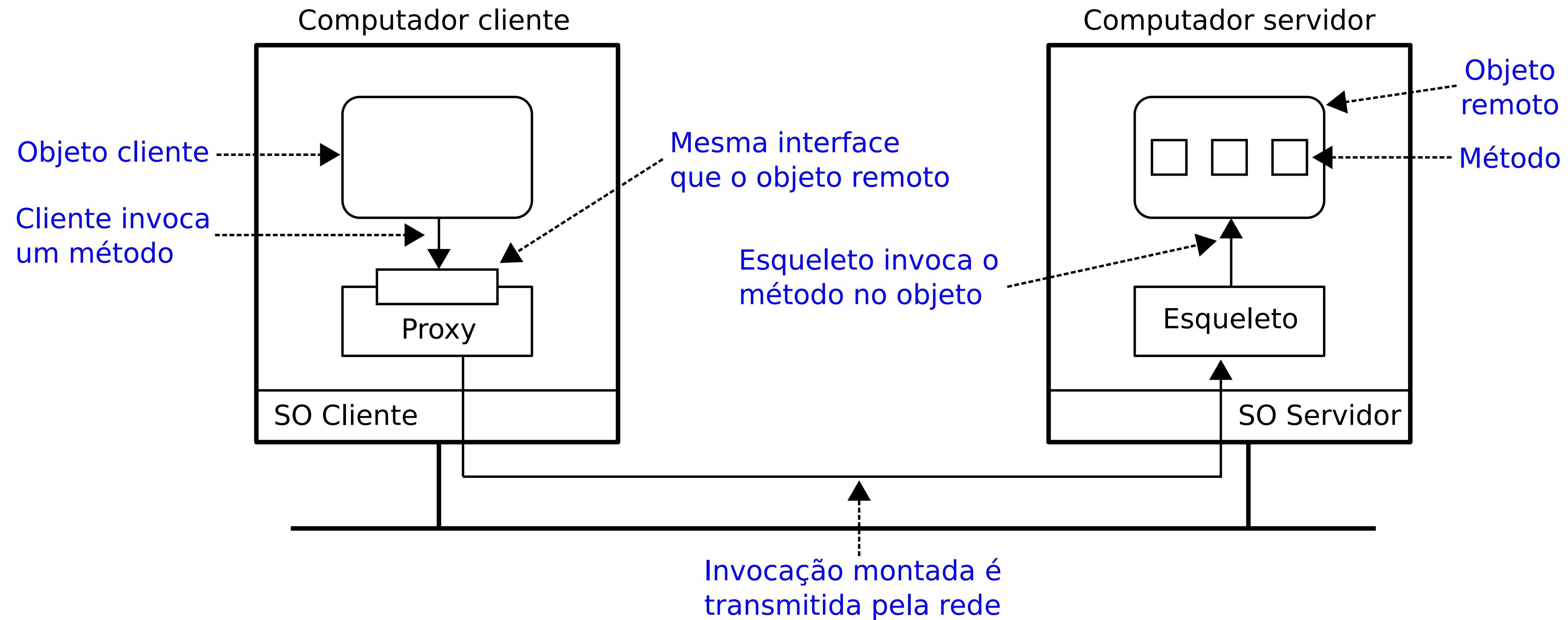
Desempacota os
argumentos

Invoca o
método
correspon-
dente

Empacota e
envia o
resultado

Invocação Remota de Métodos

Organização de Proxy e Esqueleto



Invocação Remota de Métodos

Detalhes sobre RMI

Objetos podem estar em computadores diferentes

Necessário o registro de objetos remotos

Serviço separado que mantém uma tabela de nomes textuais para referências a objetos remotos

rmi://node01:2004/HelloWorld

Necessário criar um objeto remoto e registrá-lo

Clientes criam um procurador ao objeto remoto

Cliente deve possuir uma interface remoto do objeto em questão

Invocação Remota de Métodos

Passagem de Parâmetros

Geralmente ocorre a passagem de parâmetros por valor

Considerando que um objeto remoto pode ser acessado em todo sistema distribuído

Quando um método é invocado com uma referência de um objeto remoto como parâmetro, essa referência é copiada e transferida.

Objeto local

Objeto referenciado é copiado, ou seja, é passado por valor

Objeto é passado por referência

Invocação Remota de Métodos

Desenvolvendo em RMI

1

Interfaces Remotas

Calc.java

```
import java.rmi.*;  
  
public interface Calc extends Remote {  
    int add(int i, int j) throws RemoteException;  
}
```

Invocação Remota de Métodos

Desenvolvendo em RMI

2

Programa Servidor
implementa os métodos
publicados

CalcImpl.java

```
import java.rmi.*;
import java.rmi.server.*;

public class CalcImpl extends UnicastRemoteObject implements Calc {
    public CalcImpl() throws RemoteException {
        super();
    }

    public int add(int i, int j) {
        return i + j;
    }

    public static void main(String args[]) {
        try {
            CalcImpl objRMI = new CalcImpl();
            Naming.rebind("//node01:2004/Calc", objRMI);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Invocação Remota de Métodos

Desenvolvendo em RMI

Stubs e Esqueletos gerados automaticamente

3

Programa Cliente faz uso do serviço de registro de nomes de objetos remotos

CalcClient.java

```
import java.rmi.*;  
  
public class CalcClient {  
    public static void main(String[] args) {  
        try {  
            Calc objCalc = (Calc)Naming.lookup("//node01:2004/Calc");  
            System.out.println("O resultado da soma é : " + objCalc.add(3, 7));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Invocação Remota de Métodos

Concorrência e RMI

O que acontece se vários objetos procuradores fazem acesso concorrente a métodos do objeto remoto?

Synchronized

Modificador de método

Não é possível acesso a métodos Synchronized ao mesmo tempo

Segunda e demais tentativas irão bloquear

```
public class SynchronizedCounter {  
    private int c = 0;  
  
    public synchronized void increment() {  
        c++;  
    }  
  
    public synchronized void decrement() {  
        c--;  
    }  
  
    public synchronized int value() {  
        return c;  
    }  
}
```

Invocação Remota de Métodos

Sockets TCP em Java

Cliente

```
import java.io.*;
import java.net.*;

public class Client
{
    public static void main(String[] args)
    {
        try
        {
            Socket client = new Socket("localhost", 4321);
            BufferedReader in = new BufferedReader(new
InputStreamReader(client.getInputStream()));
            PrintWriter out = new PrintWriter(client.getOutputStream(), true);
            String toSend = "I am the client";
            String toReceive;
            System.out.println("Recebi: " + in.readLine());
            out.println(toSend);
        }
        catch(Exception e)
        {
            System.out.println("Problemas em sockets: " + e);
        }
    }
}
```

Invocação Remota de Métodos

Servidor

Sockets TCP em Java

```
import java.io.*;
import java.net.*;

public class Server
{
    public static void main(String[] args)
    {
        try
        {
            ServerSocket server = new ServerSocket(4321);
            Socket client = server.accept();
            BufferedReader in = new BufferedReader(new
InputStreamReader(client.getInputStream()));
            PrintWriter out = new PrintWriter(client.getOutputStream(), true);
            String toSend = "Hello World";
            String toReceive;
            out.println(toSend);
            System.out.println("Recebi: " + in.readLine());
        }
        catch(Exception e)
        {
            System.out.println("Problemas em sockets: " + e);
        }
    }
}
```