

LEIC/LETI – 2013/14, 1º Teste de Sistemas Distribuídos, 29 de Março de 2014

Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas. Duração: 1h30m

Grupo I [7v]

1. Nas aulas teóricas foi efectuado um exercício de criar um protocolo simples de RPC usando como base um canal de comunicação TCP/UDP. A figura ao lado corresponde ao que se transmite no pacote de pedido/resposta

Identificador do pedido
Identificador da operação
Argumentos/retorno

a) [0,4] Qual a função do identificador de pedido?

b) [0,3] Em que componente do sistema de RPC será utilizado o campo “identificador de operação”?

c) [0,5] No Sun-RPC este identificador da operação remota é uma estrutura mais complexa. Que elementos permitem identificar totalmente um procedimento remoto nesse protocolo?

d) A transmissão dos argumentos e o retorno tem de ter em conta o problema da heterogeneidade.

i) [0,5] Explique em que consiste esse problema?

ii) [0,5] Indique qual a forma do SUN-RPC resolver este problema?

iii) Existem alternativas a esta solução.

(1) [0,3] Indique uma.

(2) [0,5] Indique uma vantagem e uma desvantagem da solução anterior em relação ao Sun-RPC.

2. Considere uma função remota que apenas incrementa um contador em 1 unidade e retorna o novo valor do contador. O valor deste no início é 5.

a) [0,7] Preencha a tabela de acordo com a semântica e situação de faltas indicada. Assuma cada situação isoladamente, isto é, executando-se a partir do estado inicial.

Semântica	Situação	Resultado obtido pelo cliente	Valor no Servidor
Talvez	Perde-se a mensagem de invocação		
Pelo-menos-uma – vez	Perde-se 3 vezes a mensagem de invocação		
Pelo-menos-uma – vez	A rede perde 5 vezes a mensagem de resposta do servidor para o cliente		
No- máximo-uma-vez	Perde-se 3 vezes a mensagem de invocação		
No- máximo-uma-vez	A rede perde 5 vezes a mensagem de resposta do servidor para o cliente		

b) [0,5] Explique a razão dos eventuais resultados diferentes obtidos. Seja objectivo na explicação.

3. Nos slides das aulas teóricas é apresentado este exemplo de uma função em C que representa uma transferência bancária:

```
int transfere(int origem, int destino, int valor, int *saldo, char *descr);
```

a) [0,4] Indique dois aspectos diferentes em que esta definição é ambígua em termos da execução de uma função remota.

b) [0,4] Explique como o Sun-RPC os resolve.

c) Na função aparecem apontadores para estruturas.

i) [0,3] Qual é o problema básico na sua utilização numa invocação remota?

ii) [0,5] Pode ser resolvido ou é uma limitação inultrapassável? Justifique a resposta.

d) A interface do programa remoto que executa esta função podia ser facilmente escrita na IDL do SUN-RPC.

i) [0,4] Identifique claramente os elementos da interface remota que não estão presentes na assinatura do método apresentada acima e teria de definir. Qual a sua função?

ii) [0,4] Indique quais os parâmetros que considera de entrada e quais os de saída. Explique como tal é representado no IDL.

iii) [0,4] Escreva a interface do programa completa.

Grupo II [6v]

Considere o seguinte programa (incompleto) em Java.

```

IService s;
try {

    IOtherService r = s.metodoX();
    r.metodoY();
} catch (RemoteException e) {...}
    
```

1. [0,7] Complete o programa acima para que o método seja chamado sobre um objeto remoto que foi registado com o nome "sdObj" num RMI Registry a correr na máquina "sd.tecnico.ulisboa.pt" (no porto habitual). Simplificação: omita do seu programa a definição do *SecurityManager*.

2. [0,7] Defina a interface *IService*, assumindo que só tem um método, *metodoX*.

3. [0,9] Quando o método *metodoX* é chamado, há várias componentes da arquitectura do Java RMI que estão envolvidas. Indique-as passo a passo.

	Passo	Componente envolvida
Cliente	Determinar a referência remota a enviar no pedido	
	Criar e preencher a mensagem de pedido	
	Enviar da mensagem de pedido	

Servidor	Receber a mensagem de pedido	
	Determinar o objeto local	
	Extrair os argumentos do pedido e chamar método local	
	Executar o método	

4. O conceito de proxy é fundamental para ocultar os detalhes acima ao programador, oferecendo-lhe a abstração da chamada de um método.

a. [0,6] Quantos proxies existem quando se executa o programa acima até chegar à linha "IOtherService r = s.metodoX()"? Justifique.

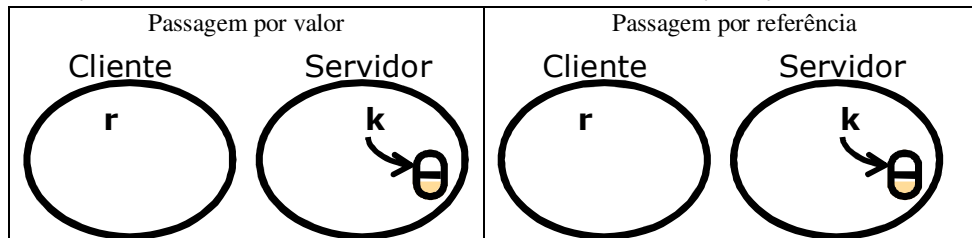
b. [0,6] Como foi/foram instanciado(s)?

5. O retorno de metodoX pode ser passado por referência ou por valor.

a. [0,7] Assuma que a última instrução do método metodoX é "return k;".

Complete as figuras seguintes por forma a ilustrar o estado após a linha "IOtherService r = s.metodoX()", consoante o retorno seja por valor ou por referência.

A sua resposta deve incluir novas instâncias e novas referências que sejam criadas (caso se aplique).



Justifique sucintamente os novos elementos que acrescentou ao desenho acima.

b. [0,6] No caso do retorno ser por valor, pode ser necessário carregar dinamicamente classes. Exemplifique descrevendo claramente uma situação em que tal ocorra.

6. O Java RMI suporta recolha automática de memória (*garbage collection*).

a. [0,6] Apresente uma instrução Java que, se acrescentada ao programa acima, causaria a chamada *removeRef*.

--

b. [0,6] Se, em vez de se usar o método de contagem de referências, se usassem leases, o que mudaria quando se executasse a instrução que indicou na alínea anterior?

Grupo III [7v]

1. A tecnologia dos Web Services (WS) teve, na sua génese, como principal objectivo a utilização da Internet, em particular da tecnologia que a WWW tinha criado. Podemos, portanto, com base nesta característica comparar a sua evolução com a tecnologia do RMI.

a) Protocolos de transporte da informação

i) [0,3] Qual a evolução principal do RMI para os WS?

ii) [0,4] Justifique a razão desta evolução.

b) Tratamento da heterogeneidade

i) [0,3] Qual a evolução do RMI para os WS?

ii) [0,4] Qual a razão principal desta evolução?

c) Tratamento das referências remotas

i) [0,4] Explique qual o tratamento que o cliente do RMI faz quando recebe uma referência remota?

ii) [0,3] Como se referencia em WS remotamente um serviço?

iii) [0,4] Explique claramente a razão porque as referências remotas são diferentes em RMI e Web Services.

POST /ExemploHelloWS/endpoint HTTP/1.1
Host: www.server.com
Content-Type: text/xml; charset="utf-8"
Content-Length: 322
SOAPAction: ""

2. Considere a figura ao lado, que mostra dois protocolos.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:ns1="http://hello">  
  <soapenv:Body>  
    <ns1:sayHello>  
      <ns1:name>friend</ns1:name>  
    </ns1:sayHello>  
  </soapenv:Body>  
</soapenv:Envelope>
```

a) [0,4] Quais? Onde está a separação entre ambos?

b) [0,4] Em que secção do WSDL foi definido que estes dois protocolos são utilizados em conjunto?

--

c) [0,5] Poderá deste exemplo saber o tipo do parâmetro "name"? Como poderá então ser garantido o tratamento da heterogeneidade? Justifique objectivamente.

d) [0,3] Qual é o URL do *endpoint* do serviço que permite a sua invocação por um cliente?

--

e) [0,3] Em que secção do WSDL deverá ter sido definido?

--

Inbound SOAP message:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Header/><S:Body><ns2:play  
  xmlns:ns2="http://tvt/"><arg0>0</arg0><arg1>1</arg1><arg2>1</arg2></ns2:play></S:Body></S:Envelope>
```

Outbound SOAP message:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><ns2:playResponse  
  xmlns:ns2="http://tvt/"><return>true</return></ns2:playResponse></S:Body></S:Envelope>
```

3. As linhas acima mostram as mensagens SOAP na invocação do exemplo do jogo que colocou em funcionamento nas aulas práticas.

a) [0,5] Escreva a mensagem de pedido com a formatação e indentação habitual em XML. Por simplificação, omita os atributos de cada *tag* na sua resposta.

b) Uma importante característica no entendimento do XML é a utilização dos *namespaces*.

i) [0,4] O que é e para que serve um *namespace*?

ii) [0,3] Um *namespace* é na sua essência um URL ou um URN? Justifique.

c) Identifique na mensagem *inbound* os *namespaces* aí presentes.

i) [0,4] Qual a razão de serem dois?

ii) [0,4] Qual a respectiva função no contexto desta mensagem?

iii) [0,6] No documento poderia ter: S:play, ns2: play, play.

Referem-se todos à mesma entidade? Justifique explicando os contextos de utilização.
