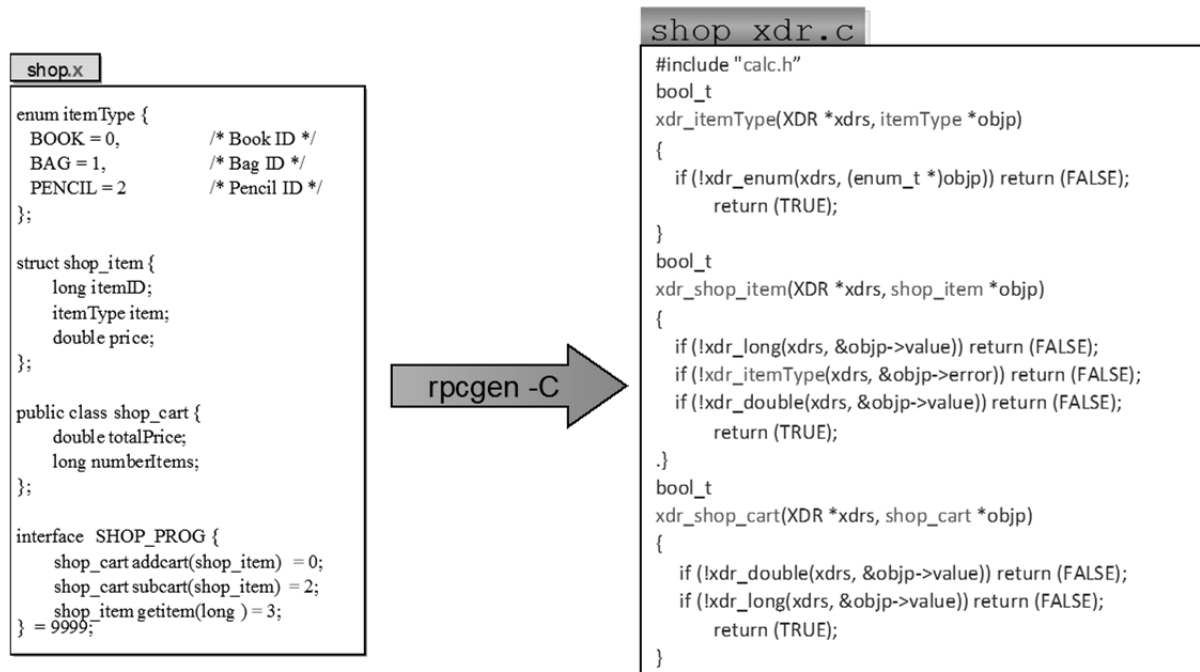


LEIC/LERC – 2011/12, 1º Teste de Sistemas Distribuídos, 13 de Abril de 2012

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

Grupo I [4,0 v]

- 1) Considere o seguinte código de uma aplicação cliente-servidor programado em Sun RPC.



- a) [0,3 v] Analise a IDL da figura para a interface de um serviço de loja. A IDL está construída de forma correta (selecione com uma cruz)?

Justifique de forma sucinta.

SIM

NÃO

- b) [0,5 v] Considere o ficheiro
- shop_xdr.c*
- da figura, e selecione com uma cruz a resposta CORRECTA (resposta errada desconta 20% da cotação desta questão):


- ☐ *rpcgen* é um compilador C que gera um programa executável a partir do código fonte em C.
- ☐ o documento define funções de conversão para cada tipo definido no IDL
- ☐ são efectuadas chamadas a funções de conversão de tipos base, como por exemplo a função *xdr_itemType*, oferecidas na biblioteca run-time do SUN RPC
- ☐ o compilador C, tanto do cliente como do servidor, irá processar ambos os ficheiros *shop_xdr.c* e *shop.x*
- ☐ Nenhuma das anteriores

2) Considere o código incompleto de um cliente SUN-RPC correspondente ao IDL da figura anterior:

```
void main (int argc, char *argv[]){
    CLIENT *___;
    int a, *result;
    char* server;
    if (argc < 2) {
        fprintf(stderr, "Modo de Utilização: %s máquina do
            servidor\n", argv[0]);
        exit(1);
    }
    server = argv[1];
    cl = clnt_create(server, _____, SHOP_VERS, "tcp");
    if(cl == NULL) {
        clnt_pcreateerror(server);
        exit(1);
    }
    ...
    shop_item sItem= getitem(10110,___);
}
```

- a) [0,5 v] Complete o código preenchendo os espaços indicados na figura.
- b) [0,5 v] `clnt_create` retorna uma estrutura. Descreva o seu conteúdo e a sua utilidade.

- c) [0,7 v] Assuma agora que se pretende que o cliente envie alternadamente o pedido *getitem* para dois servidores. Reescreva o programa no espaço em baixo.



3) Relativamente à execução distribuída de uma chamada remota:

- a) [0,5 v] Considere a seguinte semântica de execução: O run time do RPC do lado Cliente envia cada pedido apenas uma vez para o socket TCP e bloqueia-se à espera da resposta. Se o socket for fechado antes da

resposta chegar, o RPC retorna erro à aplicação. Assuma que todas as falhas de comunicação são toleradas pelo socket TCP. Esta semântica é exactamente-uma-vez? Justifique.

- b) [0,5 v] O procedimento remoto ADDCART se invocado n vezes com o mesmo item adiciona-o o mesmo número de vezes ao carrinho de compras. Poderia usar uma semântica de envio pelo-menos-uma-vez? Justifique sucintamente.

- c) [0,5 v] Considere agora a implementação do procedimento remoto GETITEM:

```
Shop_item getitem(long item){ return itemArray[item]; }
```

Poderia usar uma semântica de envio pelo-menos-uma-vez? Justifique sucintamente.

Grupo II [6,0 v]

Considere as seguintes declarações de interfaces em Java:

```
1 public interface Event extends Remote {
2     void setVenue(String venue) throws RemoteException;
3     EventObject getAllInfo() throws RemoteException;
4 }

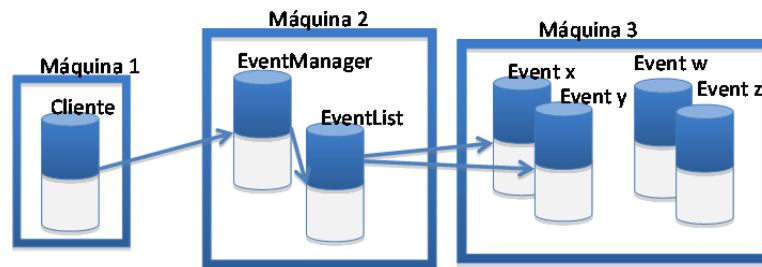
5 public interface EventList extends Serializable{ // herda de Serializable mas não de Remote
6     Event getFirstEvent(); // retorna o Event mais antigo da lista de Event
7     void addEvent(String eventName); // adiciona Event cujo nome está registado
8                                     // como EventName ao vetor de Event
9     void addNewEvent(Event e); // adiciona Event ao vetor de Event
10    Vector allEvents(); // retorna vetor de todos os Event
11 }

12 public interface EventManager extends Remote {
13     Event getFirstEvent() throws RemoteException; // retorna primeiro Event que foi inserido na
14                                                    // lista invocando getFirstEvent de EventList
15     void addEvent2List(Event e); // adiciona um Event ao vetor de events no objeto EventList
16     EventList getEvents() throws RemoteException; // retorna EventList associado a EventManager
17     int getVersion() throws RemoteException;
18 }
```

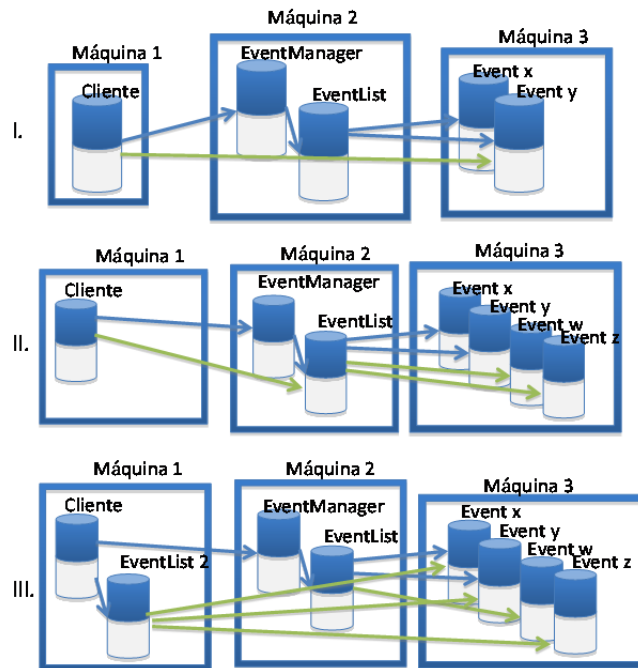
- 1) Considere a seguinte figura, onde:

- uma seta entre máquinas diferentes (de um objeto A numa máquina para um objeto B noutra máquina) significa que o objeto A detém uma referência remota para o objeto B;
- uma seta entre objetos na mesma máquina (de um objeto A para um objeto B) representa que o objeto A detém uma referência local para o objeto B.

A figura representa uma situação num dado instante de tempo, que vamos considerar o instante inicial.



- a) [0,5 v] Com base nas interfaces Java descritas e neste estado inicial dado pela figura, o cliente efetua diversas invocações usando estas interfaces, levando o sistema a um novo estado. Indique, assinalando com um círculo, qual (ou quais) das seguintes figuras (I., II. ou III.) representam um novo estado impossível para o sistema.



- b) [0,8 v] Para cada uma das alíneas anteriores:

- Para as que **não assinalou**, indique porque constitui um estado admissível, indicando como poderiam ter sido obtidas em cada máquina as referências remotas ou locais indicadas.
- Para cada uma das que **assinalou**, indique porque constitui um estado impossível.

Justifique as suas respostas.

I.

II.

III.

--

2) Na linha 14 do extrato de código apresentado anteriormente, define-se:

```
EventList getEvents() throws RemoteException;
```

Considere que, no cliente, existe uma referência remota chamada *eventManager* para uma instância remota de *EventManager*. Suponha que antes da invocação a *getEvents()* já haviam sido criadas instâncias de *Event* na máquina 3 e adicionadas 2 referências destas à *eventList* na máquina 2

Assuma também que o cliente invoca:

```
EventList eventList2=eventManager.getEvents();
```

a) [0,5 v] *eventList2* é retornado por referência ou por valor? Justifique a sua resposta.

b) i. [0,6 v] Que objetos deverão ser criados no lado do cliente após a invocação? Justifique.

ii. [0,4 v] Quantos? Justifique.

c) [0,5 v] Por que razão o método *getEvents()* lança a exceção indicada? Justifique.

d) O protocolo de recolha automática de memória (*garbage collection*) usado pelo Java RMI é do tipo de contagem de referências.

i. [0,5 v] O que terá de fazer a máquina virtual que executa o RMI do lado do cliente na sequência desta invocação em relação ao protocolo de *garbage collection*?

ii. [0,5 v] Quando esses objetos forem eliminados o que terá de fazer o cliente em relação ao protocolo de *garbage collection*?

3) Considere que após a invocação a *getEvents* da questão anterior, o cliente executa o seguinte código:

```
1 - eventList2.getFirstEvent().setVenue("Lisboa");
```

```
2 - eventManager.getFirstEvent().setVenue("Rio de Janeiro");
```

```
3 - eventList2.addEvent("SuperRock");
```

- a) [0,6 v] No final do passo 2 da execução, o evento mais antigo de eventList2 fica com que valor? Justifique.

- b) [0,6 v] Considere agora o passo 3 da execução. A alteração produzida reflete-se na lista de eventos da máquina 2? Justifique.

- c) [0,5 v] Na linha 7 do extrato de código: `void addEvent(String eventName);`
Diga a partir do conhecimento do nome da instância remota (eventName) e do endereço DNS do host onde esta está instanciada, qual o serviço que a implementação deste método poderá usar para obter uma referência remota registada nesse serviço. Justifique sucintamente.

Grupo III [5,5 v]

- 1) Os Web Services são uma evolução dos protocolos de RPC, RMI e Web Services, com diversos requisitos comuns, por vezes implementados com soluções tecnológicas diferentes.

- a) [0,5 v] Tratamento de heterogeneidade entre nós: identifique as características de cada protocolo, indicando “sim” ou “não” em cada coluna).

	Receptor-converte	Codificação Explícita
Sun RPC		
CORBA - RMI		
Web Services		

- b) [0,5 v] Justifique as escolhas no caso dos Web Services.

- c) [0,5 v] Semântica da execução de procedimento. Selecione a que cada protocolo de RPC disponibiliza.

	Talvez	Pelo menos uma vez	No máximo uma vez	Exactamente uma vez
Sun RPC – TCP				
Sun RPC – UDP				
RMI – Java				
Web – Services / HTTP				

- d) [0,5 v] Justifique a escolha no caso dos web services.

--

e) [0,5 v] Desempenho. Classificar com 1 (mais rápido) a 4 (menos rápido):

Sun RPC – UDP Web – Services RMI – Java Sun RPC – TCP

2) Considere que pretende selecionar uma tecnologia para um projeto com as seguintes características:

"A empresa XYZ quer disponibilizar eletronicamente aos seus atuais e futuros clientes vários dos seus procedimentos de encomenda. A empresa tem uma infraestrutura de IT segura protegida por uma firewall."

a) [0,5 v] Que tecnologia de invocação remota escolheria para permitir a invocação distribuída dos serviços?

b) [0,5 v] Indique duas razões fundamentais para justificar a sua resposta.

3) Considere o seguinte documento WSDL:

```
<?xml version="1.0"?>
<definitions name="StockQuote"
targetNamespace="http://example.com/stockquote.wsdl"
xmlns:tns="http://example.com/stockquote.wsdl"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
xmlns:xsd1="http://example.com/stockquote/schema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns="http://schemas.xmlsoap.org/wsdl/">

<types>
<schema targetNamespace="http://example.com/stockquote/schema"
xmlns="http://www.w3.org/2000/10/XMLSchema">
<complexType name="TimePeriod">
<all>
<element name="startTime" type="xsd:timeInstant"/>
<element name="endTime" type="xsd:timeInstant"/>
</all>
</complexType>
<complexType name="ArrayOfFloat">
<complexContent>
<restriction base="soapenc:Array">
<attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:float[]"/>
</restriction>
</complexContent>
</complexType>
</schema>
</types>

<message name="GetTradePricesInput">
<part name="tickerSymbol" element="xsd:string"/>
<part name="timePeriod" element="xsd1:TimePeriod"/>
</message>

<message name="GetTradePricesOutput">
<part name="result" type="xsd1:ArrayOfFloat"/>
<part name="frequency" type="xsd:float"/>
</message>

<portType name="StockQuotePortType">
```

```
<operation name="GetLastTradePrice" parameterOrder="tickerSymbol timePeriod frequency">
  <input message="tns:GetTradePricesInput"/>
  <output message="tns:GetTradePricesOutput"/>
</operation>
</portType>

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetTradePrices">
    <soap:operation soapAction="http://example.com/GetTradePrices"/>
    <input>
      <soap:body use="encoded" namespace="http://example.com/stockquote"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="http://example.com/stockquote"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
</definitions>
```

a) [0,5 v] Em termos de tecnologia XML, a que corresponde a secção types. Que vantagens daí advêm para o desenvolvimento dos Web Services?

b) [0,5 v] Escreva em Java a assinatura do método correspondente a este serviço.

--

c) [0,5 v] Preencha os campos correspondentes a um pacote SOAP de invocação deste serviço.

```
POST _____
Host: _____
Content-Type: text/xml; charset="utf-8"
Content-Length: 322
SOAPAction: " _____ "
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1=" _____ ">
  <soapenv:Body>
    <ns1: _____ >

    </ns1: _____ >
  </soapenv:Body>
</soapenv:Envelope>
```

d) [0,5 v] Suponha que o símbolo da ação (tickersymbol) cuja cotação é pedida é inexistente o que corresponde a uma exceção. Indique o que acrescentaria ao documento WSDL de acordo com a resposta anterior.

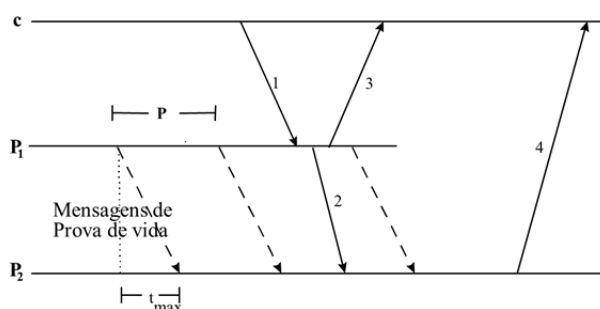
--

Grupo IV [4,5 v]

- 1) [0,75 v] As faltas nos sistemas informáticos podem ter várias características, preencha a tabela seguinte colocando para cada uma o atributo ou atributos que se lhe aplicam.

	Temporária	Permanente	Silenciosa	Arbitrária
Router descarta um pacote IP				
Cabo de alimentação de servidor desligado				
No protocolo de primary backup, o servidor primário omite uma mensagem de "I'm alive"				
Replicação ativa num sistema assíncrono, um servidor não recebe uma mensagem ao fim de x minutos				

- 2) Considere o protocolo de replicação passiva de primary-backup explicado nas aulas teóricas e ilustrado pela figura seguinte:



- a) [0,5 v] Suponha que por razões de eficiência se troca a ordem das mensagens 2 e 3, respondendo o servidor primeiro ao cliente e depois enviando a mensagem para a réplica. Esta alteração tem alguma implicação no protocolo de replicação? Justifique.

- b) [0,5 v] Suponha que o relógio do nó primário tem desfasamento crescente em relação ao relógio do secundário. Que implicação pode ter no protocolo? Como classificaria essa falta?

- c) [0,75 v] Suponha que o primário depois de enviar a mensagem ao secundário executa o serviço e envia ao cliente um resultado diferente do produzido pelo serviço. Como classifica esta situação? É tolerável? Porquê?

--

3) Considere o seguinte protocolo de replicação que se aplica a um sistema em que o cliente interatua com um serviço onde apenas atualiza (ler ou escrever) “registos”:

“O cliente envia o pedido (leitura ou escrita) a 3 réplicas e espera por um conjunto de respostas”

a) [0,5 v] Este protocolo poderia tolerar uma falta arbitrária ou bizantina. Descreva como seria o protocolo

b) [0,75 v] Explique com base no cenário inicial da pergunta o que seria um protocolo de quórum de maioria para ler um registo?

c) [0,75 v] E para escrever?
