

Número:

Nome:

## LEIC/LERC – 2010/11

### 1º Teste de Sistemas Distribuídos

9 de Abril de 2011

**Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas.**

Duração: 1h30m

#### Grupo I [5 v.]

1. [1,1 v.] Considere o seguinte IDL de um servidor baseado em SUN RPC.

---

```
enum status_t {
    OK = 0,      /* No error */
    ERROR = 1   /* Error */
};
const MAX = 1000;
typedef char name_t[MAX];

struct create_habitat_args {
    name_t zoo_name;
    name_t habitat_name;
    int area;
};

struct remove_habitat_args {
    name_t zoo_name;
    name_t habitat_name;
};

program MINI_ZOO_PROG {
    version MINI_ZOO_VERS {
        status_t CREATEZOO(name_t) = 1;
        status_t CREATEHABITAT(create_habitat_args) = 2;
        status_t REMOVEHABITAT(remove_habitat_args)=3;
    } = 1;
} = 100400;
```

---

Escreva a função main de um cliente do jogo. Assuma que tem ao seu dispor as funções `char *lerPalavra()` e `int lerInteiro()`, que lêem uma palavra e um inteiro do teclado (respectivamente). Assuma que o servidor está na máquina zoo.ist.utl.pt. O cliente deverá comunicar com o servidor por udp.

Caso necessário, tem ao dispor a seguinte função da biblioteca de run-time do SUN RPC:

`CLIENT *clnt_create(char *host, u_long prog, u_long vers, char *proto);`

2. [0,7 v] Comente a seguinte frase: “O SUN RPC, tal como os Web Services, utiliza uma representação canónica e explícita”.

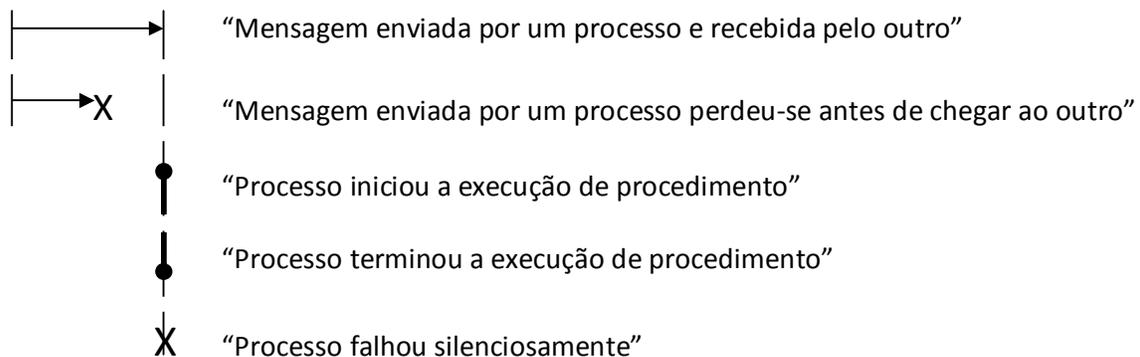
|  |
|--|
|  |
|  |
|  |

3. Considere a seguinte função remota, oferecida remotamente por algum servidor:

```
void transfere(int contaA, int contaB, int valor) {
    Saldo[contaA] -= valor;
    Saldo[contaB] += valor;
}
```

Para cada uma das situações descritas, indique (i) em quais semânticas é possível ocorrer a situação, e (ii) para uma das semânticas que indicou, apresente um exemplo de uma execução que leve a essa situação (nessa semântica).

Para ilustrar cada exemplo, utilize apenas os seguintes símbolos num diagrama temporal:



Assuma que:

- O mecanismo de comunicação usado pode perder mensagens, mas nunca causa duplicação nem reordenamento de mensagens.
- O servidor pode falhar silenciosamente e recuperar passado algum tempo.
- No estado inicial de cada alínea, ambas as contas começam com saldo 1000.
- Cada execução local da operação transfere demora 20ms, pois as alterações são aplicadas de forma persistente em disco.

- a. [0,7 v] Aplicação cliente chamou `transfere(A, B, 100)`. Recebeu “rpc-error” como resposta e estado no servidor ficou A:900 e B:1100.

Semânticas possíveis: Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

|  |
|--|
|  |
|--|

Semântica a que o exemplo se refere (escolha apenas uma, mesmo que haja múltiplas):

Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

- b. [0,7 v] Aplicação cliente chamou transfere(A, B, 100). O primeiro pedido perdeu-se na rede. No entanto, o cliente acabou por receber resposta correcta e estado no servidor ficou A:900 e B:1100.

Semânticas possíveis: Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

Semântica a que o exemplo se refere (escolha apenas uma, mesmo que haja múltiplas):

Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

- c. [0,7 v] Aplicação cliente chamou transfere(A, B, 100). Recebeu resposta correcta e estado no servidor ficou A:800 e B:1200.

Semânticas possíveis: Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

Semântica a que o exemplo se refere (escolha apenas uma, mesmo que haja múltiplas):

Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

- d. [0,7 v] Aplicação cliente chamou transfere(A, B, 100). Recebeu resposta correcta, e estado no servidor ficou A:900 e B:1100. Após receber o pedido, o servidor demorou apenas 1ms a responder com a resposta.

Semânticas possíveis: Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

Semântica a que o exemplo se refere (escolha apenas uma, mesmo que haja múltiplas):

Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

- e. [0,7 v] Aplicação cliente chamou `transfere(A, B, 100)`. Recebeu resposta correcta e estado no servidor ficou A:800 e B:1100.

Semânticas possíveis: Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

Semântica a que o exemplo se refere (escolha apenas uma, mesmo que haja múltiplas):

Talvez ; Pelo-menos-1-vez ; No-máx-1-vez ; Exactamente-1-vez

## Grupo II [5,0 v]

1. A figura indica um ficheiro de uma implementação de WebServices em que se define a classe Java, anota-se esta para gerar um Web Service, e finalmente o WSDL e os ties são gerados.

```
@javax.jws.WebService
public class CalcServiceImpl {
    @javax.jws.WebMethod
    public int intdiv(int a, int b) {
        return((int)(a / b));
    }
}
```

- a) [1,4 v] A abordagem indicada é chamada `implementation-first`. Construa a parte abstracta do WSDL gerado a partir desta implementação.

- b) [1,0 v] Considere que, para a alínea anterior, se pretende duas concretizações da interface abstracta: sobre transporte por HTTP e por SMTP. Isso seria possível no mesmo documento WSDL? Se não, justifique. Se sim, indique as secções que teriam que constar no WSDL.

|  |
|--|
|  |
|  |
|  |
|  |
|  |

- c) [1,0] Existe a possibilidade de a invocação do método remoto intdiv falhar porque o divisor é zero. Pretende-se ter uma forma de o assinalar ao cliente sem alterar os parâmetros de entrada e saída do método. Que alterações teria que efectuar no WSDL?

|  |
|--|
|  |
|  |
|  |
|  |
|  |

2. A mensagem SOAP em seguida é enviada pelo cliente durante uma invocação ao método intdiv do Webservice.

```
POST /Intdiv HTTP/1.1
Host: www.intdivserver.com
Content-Type: text/xml; charset="utf-8"
SOAPAction: "Some-URI"
```

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:intdiv xmlns:m="Some-URI">
      <m:a>2</m:a>
      <m:b>1</m:b>
    </m:intdiv>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Considere que o cliente deverá colocar um carimbo temporal em todas as mensagens enviadas ao servidor.

- a) [0,8 v] Diga que mecanismo deveria utilizar para interceptar e alterar as mensagens de forma a não introduzir alterações no lógica de negócio do serviço.

|  |
|--|
|  |
|  |
|  |

- b) [0,8 v] Colocaria o carimbo temporal em que parte desta mensagem SOAP? Justifique, indicando as alterações à mensagem SOAP que chegaria ao servidor, relativamente à mensagem original apresentada acima.

|  |
|--|
|  |
|  |
|  |

**Grupo III [5,0 v]**

```

public interface Human extends Remote {
    void setBI(int BI) throws RemoteException;
    int getBI() throws RemoteException;
    // getHealth obtem os valores de todas as propriedades do objecto da class HealthObject
    HealthObject getHealth() throws RemoteException;
}

public interface Population extends Remote {
    //HealthObject and Vector are classes whose interface extends Serializable but not Remote
    Human birth(HealthObject h) throws RemoteException;
    Vector populationHealth() throws RemoteException;
    int getSize() throws RemoteException;
}

public interface Family extends Remote {
    //Vector is classes whose interface extends Serializable but not Remote
    int addMember(Human h) throws RemoteException;
    HealthObject olderMember() throws RemoteException;
    int getSize() throws RemoteException;
}

public class FamilyServant extends UnicastRemoteObject implements Family{
    private Vector theList;
    private int familySize;

    public FamilyServant() throws RemoteException{
        theList = new Vector();
        familySize = 0;
    }
    public int addMember(Human h) throws RemoteException{
        familySize ++;
        theList.addElement(h);
        return familySize;
    }
    public HealthObject olderMember() throws RemoteException{
        return theList.firstElement().getHealth();
    }
    ...
}

public class PopulationServant extends UnicastRemoteObject implements Population{
    private Vector theList;
    private int populationSize;

    public PopulationServant() throws RemoteException{
        theList = new Vector();
        populationSize = 0;
    }
    public Human birth(HealthObject h) throws RemoteException{
        populationSize ++;
        Human s = new HumanServant(h, populationSize);
        theList.addElement(s);
        return s;
    }
    public Vector populationHealth() throws RemoteException{
        return theList;
    }
    ...
}

```

1. Considere as interfaces definidas acima e o código das 2 classes servant que implementam as interfaces Population e Family. HealthObject implementa a interface Serializable e não implementa a interface Remote. Considere ainda a seguinte sucessão de eventos, que envolvem um Servidor S1, um servidor/cliente SC1 e 4 clientes C1, C2, C3 e C4.

1. O servidor S1 cria uma instância de *PopulationServant* e regista-a no servidor de nomes com o nome rmi://S1home/MyPopulation.
2. O cliente/servidor SC1 cria uma instância de *FamilyServant* e regista-a no servidor de nomes com o nome rmi://SC1home/MyFamily.
3. Cada cliente C1, C2 e C4 obtém do servidor de nomes uma referência P1 para o objecto rmi://S1home/MyPopulation.
4. Cada cliente C1, C2 e C3 obtém do servidor de nomes uma referência F1 para o objecto rmi://SC1home/MyFamily.
5. O cliente C1 cria um objecto H1 do tipo *HealthObject* e invoca humanChild1 = P1.birth (H1).
6. O cliente C1 invoca sizeFamily=F1.addMember(humanChild1).
7. O cliente C2 cria um objecto H2 do tipo *HealthObject* e invoca humanChild2 = P1.birth (H2).
8. O cliente C2 invoca sizeFamily=F1.addMember(humanChild2).
9. O cliente C3 invoca olderM=F1.olderMember()
10. O cliente C4 invoca MyVectP=P1.populationHealth()

a) [1,3 v] No final das etapas indicadas, quantas referências remotas têm os clientes indicados? (Sugestão: use uma folha de rascunho para preparar a resposta.)

| ETAPA | SC1 | C2 | C3 | C4 |
|-------|-----|----|----|----|
| 3     |     |    |    |    |
| 4     |     |    |    |    |
| 6     |     |    |    |    |
| 8     |     |    |    |    |
| 10    |     |    |    |    |

b) Na etapa 3, vários clientes invocam o serviço de nomes. Considere o cliente C1

1. [0,5] Como resultado dessa invocação é instanciado um objecto no espaço de endereçamento do cliente C1. Qual? Justifique a resposta

|  |
|--|
|  |
|  |
|  |

2. [0,7] Esse objecto tem uma classe associada. Como sabe o run-time do cliente qual a classe a carregar?

|  |
|--|
|  |
|  |
|  |

c) [0,5 v] Qual o valor devolvido pelo método *addMember* na etapa 8? Justifique.

|  |
|--|
|  |
|  |
|  |
|  |

d) [0,6 v] Considere que numa futura etapa 11, C4 invoca *MyVectP.lastElement().setBI(123)* para actualizar o BI de 222 para 123. De seguida na etapa 12, C2 invoca *humanChild2.getBI()*. Qual o valor de BI para *humanChild2* obtido por C2? Justifique.

|  |
|--|
|  |
|  |
|  |

2. Admita que o garbage collector utiliza Contagem por Referências.

a) [0,9 v] Desde a etapa 3 ao fim da etapa 10, preencha a tabela, indicando qual o valor do contador para os objectos *MyFamily* e *humanChild2* e qual o servidor que aloja a instância *servant* destes objectos. Indique também em que etapa o contador é criado. Exclua referências locais na sua resposta.

| Objectos    | Servidor | Etapas | Contador |
|-------------|----------|--------|----------|
| MyFamily    |          | 3      |          |
|             |          | 4      |          |
|             |          | 5      |          |
|             |          | 6      |          |
|             |          | 7      |          |
|             |          | 8      |          |
|             |          | 9      |          |
|             |          | 10     |          |
| humanChild2 |          | 3      |          |
|             |          | 4      |          |
|             |          | 5      |          |
|             |          | 6      |          |
|             |          | 7      |          |
|             |          | 8      |          |
|             |          | 9      |          |
|             |          | 10     |          |

b) [0,5 v] Considere agora que o cliente C4 não precisa mais de aceder a *humanChild2*, pelo que a *Proxy* local é destruída pelo *Garbage Collector* de C4. Que alteração, se alguma, introduz este passo no contador de referências no servidor para *humanChild2*?

|  |
|--|
|  |
|  |

## Grupo IV [5 v]

Considere a seguinte protocolo de replicação activa.

Assuma um sistema com **N=7 réplicas** de um dado registo.

Cada réplica mantém um par <valor, (numSeq, idUltimoClienteEscrivor)>.

O sistema replicado oferece aos clientes uma interface com duas operações: leitura e escrita.

Cada operação implica os seguintes passos:

---

ler() {

1. cliente envia pedido de leitura a todas as réplicas;
2. cada réplica que recebe o pedido responde com <valor,(numSeq, idUltimoClienteEscrivor)>;
3. cliente bloqueia-se à espera pela resposta de **R** réplicas;
4. cliente selecciona a resposta com maior (numSeq, idUltimoClienteEscrivor) e retorna o respectivo valor à aplicação;

}

escrever(novoValor) {

1. cliente envia pedido de leitura a todas réplicas;
2. cada réplica que recebe o pedido responde com <valor,(numSeq, idUltimoClienteEscrivor)>;
3. cliente bloqueia-se à espera pela resposta de **R** réplicas;
4. cliente selecciona  $\text{maxSeq} = \max(\text{numSeq})$  de entre as respostas recebidas;
5. cliente envia pedido de escrita <novoValor,(maxSeq +1,idCliente)> a todos as réplicas;
6. cada réplica que receba o pedido de escrita aplica-o caso o timestamp no pedido de escrita seja superior ao timestamp (numSeq, idUltimoClienteEscrivor) actual da réplica e responde *ack*; caso contrário, não aplica a escrita e responde *ack*;
7. cliente bloqueia-se até que todas as respostas *ack* chegue de **W** réplicas;
8. cliente retorna à aplicação;

}

- 
1. Considere que **R=4 e W=4**.

Para cada afirmação seguinte, indique se é verdadeira ou falsa, justificando sucintamente. No caso de afirmações falsas, justifique com exemplo de uma execução que refute a afirmação.

- a. [1 v] “O protocolo tolera a falta simultânea de 4 réplicas.”

|  |
|--|
|  |
|  |
|  |
|  |
|  |

- b. [1 v] “Uma leitura que seja pedida sem que haja qualquer outra operação (leitura/escrita) a ocorrer em concorrência irá garantidamente retornar o último valor escrito pela escrita mais recente”.

|  |
|--|
|  |
|  |
|  |
|  |

|  |
|--|
|  |
|--|

- c. [1 v] “Uma leitura  $r$  que seja iniciada depois de uma escrita  $w$  iniciar, mas antes de  $w$  terminar, vai necessariamente retornar o valor escrito por  $w$ .”

|  |
|--|
|  |
|  |
|  |
|  |
|  |

2. [1 v] Considere agora que **R=2 e W=6**.

O sistema continuaria correcto? Se sim, justifique. Se não, ilustre com um exemplo.

|  |
|--|
|  |
|  |
|  |
|  |
|  |

3. [1 v] Assuma que ambas as soluções são correctas (independentemente da resposta que deu à alínea anterior). Indique um cenário em que  $R=4/W=4$  é mais vantajoso, e outro cenário em que  $R=2/W=6$  é mais vantajoso.

|  |
|--|
|  |
|  |
|  |
|  |