Número: Nome:

LEIC/LERC – 2010/11

2º Exame de Sistemas Distribuídos

24 de Junho de 2011

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

Duração: 2h30m

Grupo I [2,5v]

Em Unix, a macro *clnt_call* permite a programas escritos em C invocar procedimentos oferecidos em servidores remotos (por exemplo, através do SUN RPC ou outros RPCs). A descrição das man pages é a seguinte:

int clnt_call(CLIENT *clnt, u_long procnum, xdrproc_t inproc, xdrproc_t outproc, char *in, char *out, struct timeval tout)

A macro that calls the remote procedure ***procnum*** associated with the client handle, ***clnt***, which is obtained with an RPC client creation routine such as clnt_create(). The parameter ***in*** is the address of the procedure's argument(s), and ***out*** is the address of where to place the result(s); ***inproc*** is used to encode the procedure's parameters, and ***outproc*** is used to decode the procedure's results; ***tout*** is the time allowed for results to come back.

This routine **returns zero if it succeeds, or an error value (a non-null integer) if it fails**.

1. [0,4v] Como sabe qual o socket e qual o endereço do servidor?

2. [0,4v] Onde é tratada a heterogeneidade?

3. [0,4v] Se se pretendesse cifrar os parâmetros enviados num pedido, a cifra deveria acontecer antes da função *inproc* ser chamado, ou depois? Justifique.

4. [0,5v] A mensagem enviada com o pedido e a mensagem retornada com a resposta podem ser vistas como tipos estruturados, contendo múltiplos campos. Indique os campos que a mensagem de pedido e a mensagem de retorno incluem. Tal como acima, assuma que a semântica oferecida é no-máximo-uma-vez. Para simplificar, assuma um cliente único e que as mensagens são enviadas em datagramas UDP de dimensão ilimitada.

5. O IDL do DCE RPC permite que programador associe aos procedimentos remoto o atributo *idempotent*, que indica que a implementação do procedimento é idempotente.
   a. [0,4v] O que é um procedimento idempotente? Ilustre com pseudo-código da implementação de um procedimento idempotente que modifique o estado do servidor.

b. [0,4v] Que vantagem encontra tornar explícito no IDL quais os procedimentos que são idempotentes?

Grupo II [2,5v]

1. Os Web Services podem ser vistos como uma plataforma de RPC, na mesma categoria de plataformas mais antigas como o SUN RPC ou o DCE RPC.
   a. [0,6v] Complete a figura na coluna da esquerda (usando os números correspondentes ao código da coluna da direita) para indicar a descrição de uma parte do serviço na linguagem WSDL, que consiste numa operação para votar, com a seguinte assinatura: *Resultado votar(long votante, string escolha)*.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Vote"
  targetNamespace="http://exemploTesteSD.com/vote/vote"
  xmlns:tns="http://exemploTesteSD.com/vote/vote"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
```

(COLOQUE OS Nºs DA DIREITA PELA ORDEM CORRECTA)

```
•
•   <xsd:schema elementFormDefault="qualified"
    targetNamespace="http://exemploTesteSD.com/vote/vote">
•
•           <xsd:restriction base='xsd:string' >
•             <xsd:enumeration value="SUCESSO" />
•             <xsd: enumeration value="ERRO" />
•           </xsd:restriction>
•
•     </xsd:schema>
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•   </operation>
•
```

1. </message>
2. <types>
3. <message name="*voteOutMessage*">
4. <xsd:simpleType name="*Resultado*">
5. <part name="*reply*" type="*tns:Resultado*"/>
6. </portType>
7. </types>
8. <part name="*votante*" type="*xsd:long*"/>
9. </xsd:simpleType>
10. <operation name="*votar*">
11. <message name="*voteInMessage*">
12. <portType name="*VotePortType*">
13. <input message="*tns:voteInMessage*"/>
14. <output message="*tns:voteOutMessage*"/>
15. <part name="*escolha*" type="*xsd:string*"/>
16. </message>

   b. Considere que a operação votar deveria retornar uma estrutura, que para além do Resultado do processo de voto, indica também o número total de votos já efectuados pelo utilizador.
      i. [0,2v] Que secção deveria modificar?

      ii. [0,4v] Reescreva a parte estritamente necessária.

c. [0,5v] Considere que existe a possibilidade de a invocação falhar porque a escolha de voto do utilizador não existe, e que pretende ter uma forma de o assinalar ao cliente. Diga que alterações teria que introduzir.

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

a. [0,4v] Foi feito o deploy deste WebService em "xpto.ist.utl.pt/exemplo". Defina o pacote SOAP correspondente à invocação do método votar deste serviço. Assuma que os valores dos parâmetros de entrada são 1 e "sd", respectivamente.

```
POST /__________ HTTP/1.1
Host: ____________________________________
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: ""
<SOAP-ENV:Envelope
< namespaces - Não Preencher >……
   <SOAP-ENV:Body>
     <m:____________________ xmlns:m="um-URI">
     <_______>_____</_________>
     <_______>_____</_________>
     </m:___________________>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

b. [0,4v] Classifique a abordagem seguida pelo SOAP para a resolução da heterogeneidade, no que toca à estrutura das mensagens e à política de conversão dos dados. Justifique.

Grupo III [2,5v]

```
1    package examples.RMIGoods;
2    import java.rmi.*;
3    import java.rmi.server.*;
4
5    public class StuffMaterial implements Serializable{
6      public StuffMaterial() {……}
7    }
8
9    public class ShopServant extends UnicastRemoteObject implements Shop{
10    private Vector theList;
11    private int itemsNumber;
12    Producer fProducer = null;
13
14      public ShopServant() throws RemoteException{
15        theList = new Vector();
16        itemsNumber = 0;
17      }
18      public int addItem(Stuff s) throws RemoteException{
19        itemsNumber ++;
20        theList.addElement(s);
21        if (!fProducer)
22          fProducer  = (Producer) Naming.lookup("//agriculture.net/Producer");
23        long ID=s.GetID();
24        fProducer.CheckStuff(ID);
25        return itemsNumber;
26      }
27      public StuffMaterial recentItem() throws RemoteException{
28       return theList.lastElement().getStuff();
29     }
30   …
31   }
32
33   public class StuffClient{
34     public static void main(String args[]){
35           System.setSecurityManager(new RMISecurityManager());
36           Producer aProducer = null;
37           Shop aShop = null;
38           Int numberItems = 0;
39           try{
40                     aProducer = (Producer) Naming.lookup("//agriculture.net/Producer");
41                     StuffMaterial sm = new StuffMaterial();
42                     Stuff apple=aProducer.collect(sm);
43                     aShop  = (Shop) Naming.lookup("//agriculture.net/Shop");
44                     numberItems=aShop.addItem(apple);
45                     StuffMaterial sm2 = new StuffMaterial();
46                     Stuff potato=aProducer.collect(sm2);
47                     numberItems=aShop.addItem(potato).
48            }catch(RemoteException e) {System.out.println("StuffStatus: " + e.getMessage());}
49     }
50   }
51
52   public class ShopClient{
53     public static void main(String args[]){
54           System.setSecurityManager(new RMISecurityManager());
55           Shop aShop = null;
56           try{
57                     aShop  = (Shop) Naming.lookup("//agriculture.net/Shop");
58                     StuffMaterial recentS = aShop.recentItem();
59            }catch(RemoteException e) {System.out.println("ShopStatus: " + e.getMessage());}
60     }
61   }
```

Considere o seguinte extracto acima de programas que descrevem a classe StuffMaterial e as classes de clientes de uma aplicação distribuída para a gestão da distribuição de produtos agricolas de um produtor. As interfaces Stuff, Producer e Shop herdam da interface Remote. Considere ainda as instancias Producer e Shop em servidores diferentes.

1. [0,4v] A definição da interface Shop herda da classe Remote. A única notação que é necessário para um método ser invocável remotamente é a sua interface herdar de Remote. Concorda com esta afirmação? Justifique.

2. Considere as linhas 40, 42 do cliente StuffClient, e a linha 58 do ShopClient.
   a. [0,5 v] Em cada um dos casos, que informação é retornada por essas invocações? Como resultado dessas invocações são também instanciados objectos no espaço de endereçamento dos clientes. Quais? Justifique a resposta.

   b. [0,5v] Esses objectos instanciados têm uma classe associada. Indique para cada caso da alínea anterior como sabe o run-time do cliente qual a classe a carregar.

3. [0,3v] Qual o valor de numberItems no StuffClient no final? Justifique.

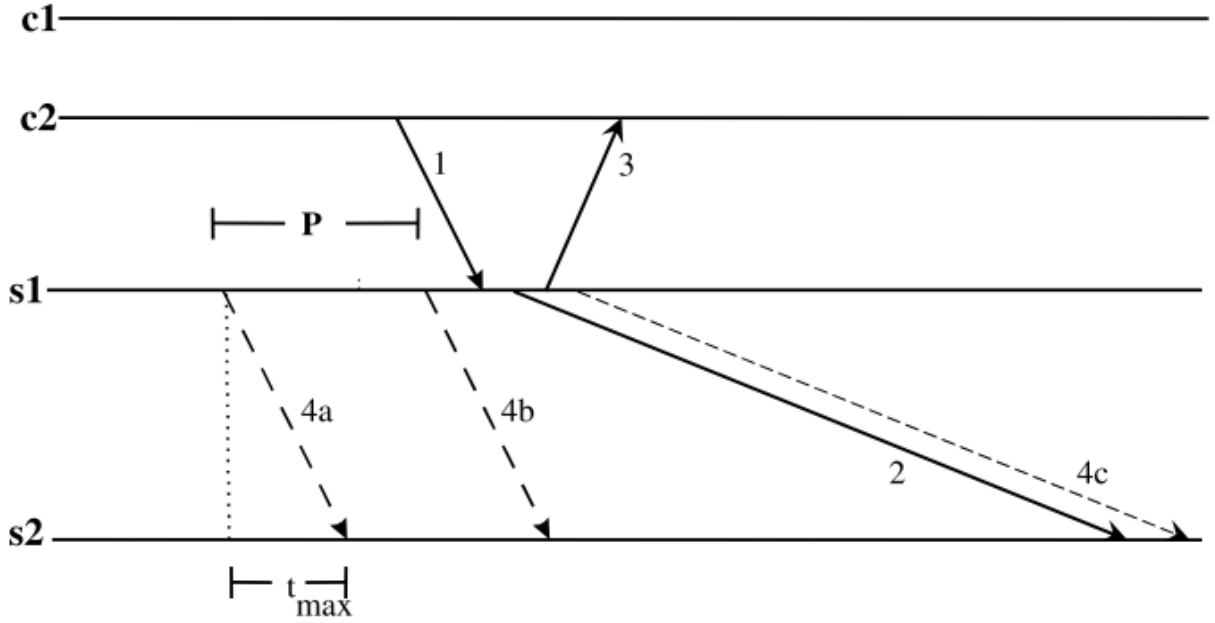
4. [0,4v] Para que servem os registos no servidor de nomes? E os `Naming.lookup` da figura?

5. [0,4v] Admita que o garbage collector utiliza leases. Considere que os clientes terminam. O que implica a terminação dos clientes relativamente ao protocolo de gestão de memória no servidor? Justifique.

Grupo IV [2,5v]

Para dotar um dado servidor s1 de maior tolerância a faltas, decidiu-se replicá-lo noutro servidor s2, usando o protocolo simples de replicação passiva *primary-backup*. Quando o protocolo foi implementado, assumiu-se que a comunicação entre s1 e s2 era síncrona, com um tempo máximo de propagação de mensagens t_{max}.
Cada cliente interage com o sistema da seguinte forma: envia pedido a um dos servidores, escolhido aleatoriamente em cada pedido; caso o servidor que recebe o pedido seja actualmente o primário, este executa o pedido e responde ao cliente; caso o servidor contactado seja o secundário, este retorna uma excepção ao cliente, que então re-envia o pedido para o outro servidor.

Considere o seguinte diagrama temporal de uma execução, em que c1 e c2 são clientes. As mensagens 4a, 4b, 4c são enviadas por s1 com intervalos de P segundos.



1. [0,5v] Preencha a legenda da figura:

1	4a
2	4b
3	4c

2. O pressuposto considerado na altura em que o protocolo foi desenhado e implementado claramente não se observou na execução ilustrada acima.
   a. [0,5v] Que pressuposto é esse? E que parte(s) da figura mostram que ele não se observa?

   b. [0,5v] Como chamaria a esta falha?

   c. [1v] Esta falha pode ter consequências graves, levando o sistema a comportar-se de forma incorrecta. Ilustre completando a figura acima (desenhe sobre a figura) com mensagens adicionais que mostrem uma execução incorrecta. Identifique cada mensagem por A, B, C, D, etc. e preencha a respectiva legenda abaixo.

Sugestão: considere que tanto o cliente c1 como o cliente c2 pretendem invocar uma operação conflituante, para a qual o sistema só deveria permitir que um dos dois clientes tivesse sucesso.

Grupo V [3v]

1. Num projecto da cadeira de Sistemas Distribuídos, pedia-se a implementação de um sistema distribuído que permitisse a clientes invocarem transferências entre diferentes servidores de contas bancárias, segundo o seguinte procedimento executado no cliente:

```
1 Boolean transferência (bancoA, bancoB, Valor)
2 {
3   idDtx = openTransaction();
4   Int saldoA = LerSaldo (bancoA, idDtx);
5   Int saldoB LerSaldo (bancoB, idDtx);
6   if (Valor > SaldoA)
7     closeTransaction(idDtx, abort);
8   else
9   {
10    ActualizarSaldo (bancoA, saldoA-Valor, idDtx);
11    ActualizarSaldo (bancoB, saldoB+Valor, idDtx);
12    return closeTransaction(idDtx, commit);
13  }
14}
```

O protocolo implementado para garantir a confirmação atómica da transacção distribuída é o 2-Phase Commit. No projecto entregue por um grupo de alunos detectaram-se vários potenciais problemas. Nas alíneas seguintes descrevem-se esses problemas. Em cada alínea, ou i) apresente um exemplo de execução que ilustre o problema, ou ii) indique que não há problema e justifique.

a. [0,6v] Quando o método closeTransaction do cliente é chamado, o cliente envia o pedido "closeTransaction" ao coordenador de forma assíncrona e retorna de imediato *true* à aplicação, sem esperar pela resposta do coordenador.

b. [0,6v] Quando o coordenador recebe o pedido "closeTransaction(idDtx, commit)", envia canCommit bancoA e bancoB e espera até 1seg pelos respectivos votos. Passado esse tempo, e mesmo que a resposta de algum dos bancos não tenha ainda chegado, verifica se os restantes votos que recebeu são todos YES. Se sim, envia decisão doCommit a ambos os bancos, incluindo aos bancos que não tenham respondido a tempo.

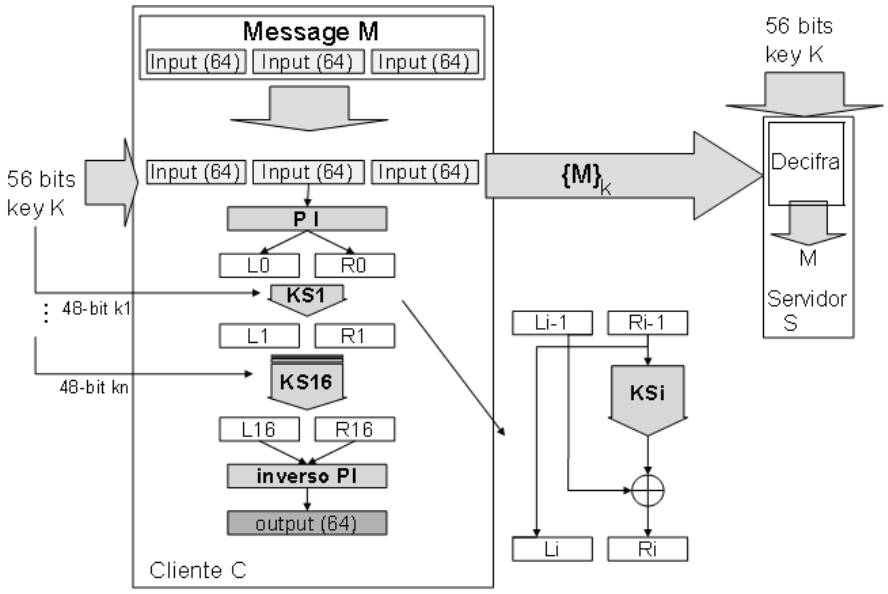
c. [0,6v] Quando um banco recebe o pedido de canCommit e responde YES, tranca completamente a sua base de dados até receber a decisão final para a transacção idDtx. Durante esse período nenhuma outra transacção distribuída será aceite nesse banco.

d. [0,6v] Se, após ter votado YES, um banco não receber a decisão do coordenador dentro de um tempo razoável, o banco decide confirmar e comunica essa decisão ao outro banco que participa na transacção.

e. [0,6v] Se, após receber o pedido canCommit, um banco votar No, esse mesmo banco aborta a transacção local de imediato, sem esperar pela decisão final do coordenador.

Grupo VI [5v]

1. Considere a figura seguinte, a qual representa a utilização de um dos algoritmos de cifra estudados nas aulas. A figura ilustra a cifra de uma mensagem e o envio desta cifrada {M}k de um cliente C para um servidor S, o qual decifra a mensagem. A mensagem M divide-se em vários "input(64)", cada qual é cifrado individualmente.



a. [0,2v] Como classifica a cifra empregada por C e S em que ambos usam a mesma chave k: cifra simétrica ou assimétrica?

b. [0,2v] Indique o nome do algoritmo de cifra que está a ser utilizado pelo Cliente C para cifrar a mensagem M? E qual o algoritmo de cifra utilizado por S para decifrar {M}k?

c. [0,5v] Usando o modo de cifra indicado em cima, poderão existir padrões na mensagem original que se mantêm na mensagem cifrada. Diga como poderia resolver esta situação.

2. Considere os mecanismos de segurança que deu nas aulas para implementar politicas de segurança.
   a. [0,5v] Descreva mecanismos de segurança que conheça para garantir as seguintes propriedades de segurança (indique apenas em cada linha o mecanismo que garanta apenas o conjunto de propriedades indicadas na coluna Propriedade da tabela). Não pode referir repetidamente o mesmo mecanismo de segurança em várias linhas.

Propriedade	Mecanismo de segurança
Confidencialidade	
Integridade	
Integridade e Autenticação	

Integridade, Autenticação e não-repudiação	
Autorização	
Distribuição segura de chaves	

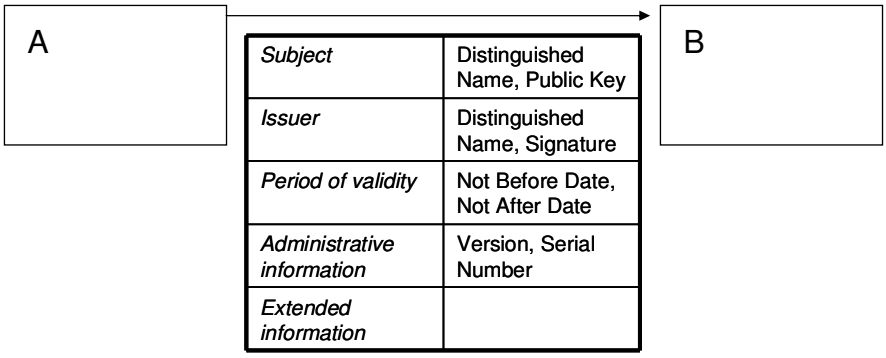
b. Pode ser usada uma função de dispersão/resumo para efectuar o resumo da mensagem M, tal que h(M) = [tamanho(M)]^3 mod 15.

I. [0,4v] Considera esta função uma função de dispersão apropriada para o resumo de mensagens, com o objectivo de assegurar a propriedade de segurança integridade? Se sim, justifique. Se não, indique que propriedade(s) ela não assegura.

II. [0,5v] Assuma que A tem duas versões de um contrato, B e M. A versão F é favorável ao utilizador B, e a versão M do contrato é má para B. Descreva um ataque possível que um utilizador A poderá efectuar a um utilizador B, decorrente da utilização por B desta função de resumo na criação de uma assinatura digital usando a chave privada de B.

3. Considere o seguinte exemplo, ilustrando o envio de um certificado X.509 da chave pública de A, de A para B. Considere que o certificado foi criado e distribuído por uma entidade de confiança

.



a. [0,4v] Qual a razão da necessidade de utilizar certificados de chave pública? Justifique.

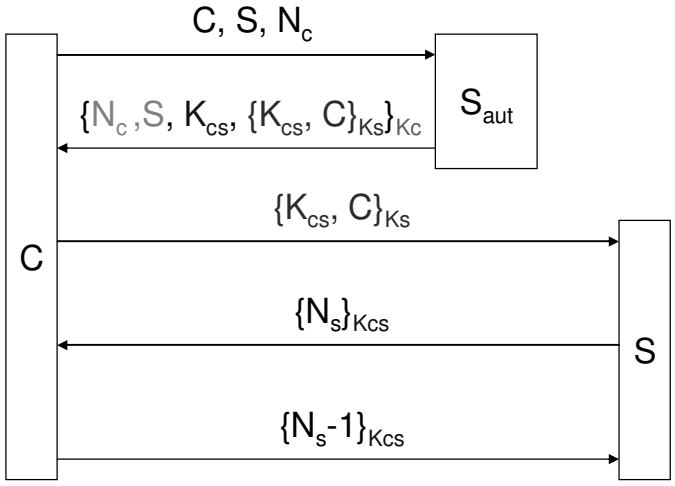
b. [0,4v] Após B receber o certificado, como é que B atesta a veracidade da informação contida neste?

c. Após B receber o certificado, verifica também que este está válido.

I. [0,4v] Como é que B atesta a validade temporal do certificado?

II. [0,5v] O certificado pode ter validade temporal, mas já não ser válido do ponto de vista da Autoridade Certificadora? Justifique.

4. Considere o Protocolo de Needham-Schroeder de criptografia simétrica.



a. [0,5v] Diga o que é Nc e para que serve esta informação no contexto da figura acima.

b. [0,5v] Este protocolo da figura é vulnerável a um ataque. Diga qual, justificando.

Grupo VII [2v]

Da Wikipedia:
A universally unique identifier (UUID) is an identifier standard used in software construction, standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE). The most widespread use of this standard is in Microsoft's globally unique identifiers (GUIDs).
A UUID is a 16-byte (128-bit) number. The number of theoretically possible UUIDs is therefore about 3 × 1038. In its canonical form, a UUID consists of 32 hexadecimal digits, displayed in 5 groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 digits and 4 hyphens). For example:
550e8400-e29b-41d4-a716-446655440000
Version 4 UUIDs use a scheme relying only on random numbers. This algorithm sets the version number as well as two reserved bits. All other bits are set using a random or pseudorandom data source. Version 4 UUIDs have the form xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx where x is any hexadecimal digit and y is one of 8, 9, A, or B.

1. [0,4v] Como classificaria os UUIDs quanto à pureza? Justifique.

2. [0,4v] E quanto à homogeneidade? Justifique.

3. [0,4v] Assuma que pretendia usar UUIDs para identificar utilizadores num sistema em que sabe, *a priori*, que só haverá um máximo de 250 utilizadores. Por essa razão, decide usar uma versão simplificada de UUIDs usando um esquema baseado em números aleatórios , em que cada identificador tem apenas 8 bits (permitindo até 256 utilizadores). Qual o problema desta solução? Justifique referindo uma propriedade dos nomes que seja posta em causa.

4. Compare os UUIDs com os nomes DNS, indicando uma vantagem de um em relação a outro, justificando:

a. [0,4v] Tendo em conta o registo de novas associações.

b. [0,4v] Tendo em conta a resolução de nomes.
