	Departamento de Informática **Faculdade de Ciências e Tecnologia** **UNIVERSIDADE NOVA DE LISBOA**

Licenciatura em Engenharia Informática
Sistemas Distribuídos I – 2ª chamada, 6 de Julho de 2005
2º Semestre, 2004/2005

Exame sem consulta com a duração de 2h30min

Aluno nº _____Nome: _____

1. Responda às seguintes perguntas.
   a. Indique qual ou quais os serviços/servidores de nomes que é necessário contactar para resolver o nome "rmi://sdserver.di.fct.unl.pt:12345/objs/xpto" (por exemplo, quando se faz "**Naming.lookup("//sdserver.di.fct.unl.pt:12345/objs/xpto")**"). Para cada serviço, indique o nome a resolver.

   b. Em Corba existe a possibilidade de definir que um tipo de um parâmetro pode ter qualquer tipo (tipo *any*). Existe algum mecanismo semelhante em Java RMI? Qual?

   c. Na transmissão de valores entre clientes e servidores, o sistema Corba não envia informação dos tipos de dados que transmite (ignore os parâmetros de tipo *any*). Compare esta aproximação com a aproximação usada por omissão no Java indicando as vantagens e desvantagens de cada uma (não se esqueça de considerar os problemas colocados por passar objectos como parâmetro).

   d. Num sistema de comunicação síncrona unidireccional, é necessário o receptor enviar alguma mensagem ao emissor? Justifique.

SIM, porque / NÃO, porque (risque o que não interessar)

e. Supondo que o sistema de comunicações não introduz falhas de duplicação nem de omissão (i.e. não perde pacotes), qual a semântica que melhor caracteriza o comportamento da execução de operações sem resultados de retorno por um sistema de RPCs que use o protocolo *request* sobre UDP: "no máximo uma vez", "exactamente uma vez" ou "pelo menos uma vez"? Justifique.

No máximo uma vez / **Exactamente uma vez** / **Pelo menos uma vez** (rique o que não interessar) **porque**

f. Num sistema de RPCs, considere o modelo em que para cada objecto existe apenas um *thread* responsável por processar as invocações remotas recebidas em sequência, processando cada invocação sem interrupção desde o início até ao fim. Indique justificadamente quais as vantagens e desvantagens desta aproximação quando comparada com um modelo em que é criado um thread para atender cada invocação remota.

2. Considere o contexto de um sistema distribuído de ficheiros semelhante ao proposto no trabalho prático, composto por um servidor de espaço de nomes e vários servidores de conteúdos. Supondo que:
   - cada cliente (CLT) partilha um chave secreta (simétrica) com o servidor de nomes (SN).
   - cada cliente (CLT) partilha um chave secreta (simétrica) com cada um dos servidores de conteúdos (SC).

   a. Apresente um protocolo que permita a um cliente executar, de forma segura, uma operação com duas partes, a primeira parte (M1) a executar no servidor de nomes e a segunda parte (M2) a executar no servidor de conteúdos e dependente do resultado da primeira parte (R1), i.e., o CLT deve enviar M1 e M2, o SN deve receber M1 e gerar R1, o SC deve receber M2 e R1 e gerar R2, e o CLT deve receber R2 – assuma que o cliente sabe, quando inicia a execução da operação, qual o SC que será contactado. O protocolo deve ter, no máximo, três mensagens, a primeira entre o CLT e o SN, a segunda entre o SN e o SC e a terceira entre o SC e o CLT. O protocolo deve garantir o secretismo das mensagens (i.e., apenas o destinatário de cada informação deve poder obtê-la) e a autenticação das mensagens (i.e., quem recebe uma informação deve poder garantir a sua origem). Explique como seriam garantidas as propriedades indicadas. Use as notações sintéticas de descrição de protocolos criptográfico que aprendeu nas aulas.

      NOTA: Caso não consiga resolver o problema assumindo que os elementos do sistema apenas conhecem a informação e chaves indicadas, indique explicitamente qual a informação e chaves adicionais que assume serem conhecidas por cada um dos elementos do sistema.

CLT -> SN:

O que garante o secretismo?

Como se garante quem gerou a informação usada (M1,…)?

SN -> SC:

O que garante o secretismo?

Como se garante quem gerou a informação usada (M2,R1,…)?

SC -> CLT:

O que garante o secretismo?

Como se garante quem gerou a informação usada (R2,…)?

Definição dos símbolos usados (complete, se necessário):
M1 – mensagem codificando a 1ª parte da operação a efectuar, criado por CLT e a entregar em SN
M2 – mensagem codificando a 2ª parte da operação a efectuar, criado por CLT e a entregar em SC
R1 – resultado da 1ª parte da operação, criado por SN e a entregar em SC
R2 – resultado da 2ª parte da operação, criado por SC e a entregar em CLT
Kn – chave simétrica partilhada entre o CLT e o SN
Kc – chave simétrica partilhada entre o CLT e o SC envolvido na execução da operação

3. No algoritmo Diffie-Hellman, dois parceiros de comunicação, Alice e Bob, trocam dois valores em claro a partir dos quais geram uma chave simétrica apenas conhecida por eles. Esta chave permite garantir o secretismo das mensagens trocadas entre ambos. De seguida apresenta-se o protocolo estabelecido entre A e B:
   A -> B : Ya
   B -> A : Yb
   A e B calculam Ks de forma independente
   A -> B : {m1}Ks
   B -> A : {m2}Ks
   A -> B : {m3}Ks
   …

a. Ignorando todos os problemas de autenticação que o protocolo anterior apresenta, indique as vantagens (se alguma) do método descrito quando comparado com a troca simples de mensagens cifradas com uma chave secreta partilhada. Justifique a sua resposta.

b. Suponha que A e B partilham dois segredos, s1 e s2. Se, após calcular Ks, A enviar a B {s1}Ks (i.e. m1=s1), e B enviar a A {s2}Ks (i.e. m2=s2), A e B conseguem autenticar-se mutuamente e garantir o secretismo das mensagens trocadas usando Ks? Justifique.

SIM, porque / NÃO, porque (risque o que não interessar)

4. Considere um sistema composto por três processos, os quais comunicam através da troca de mensagens ponto-a-ponto e multi-ponto usando um sistema de middleware que garante a fiabilidade da entrega das mensagens. Neste contexto, indique se cada uma das seguintes afirmações é **[V]erdadeira** ou **[F]alsa**. **Nota: as respostas erradas descontam.**

i. Dados dois eventos, e1 e e2, ocorridos no mesmo processo e estampilhados com um relógio lógico de Lamport, C(e1)=5 e C(e2)=8, sabe-se que e1 aconteceu antes de e2.

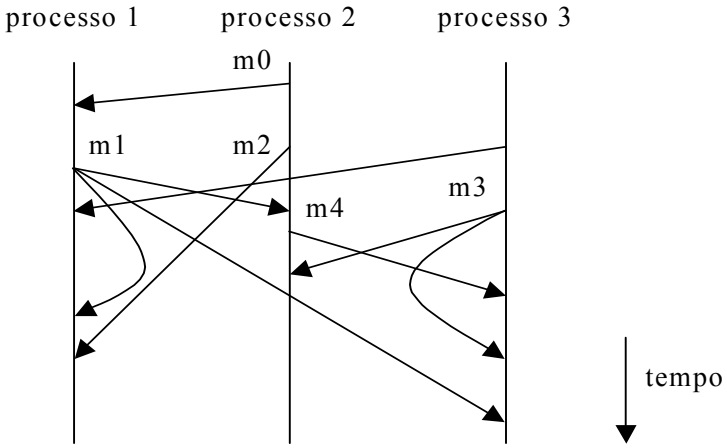
ii. Dados dois eventos, e1 e e2, ocorridos em processo diferentes e estampilhados com um relógio lógico de Lamport, C(e1)=5 e C(e2)=8, sabe-se que e1 aconteceu antes de e2.

iii. Dados dois eventos de envio de mensagem multi-ponto, e1 e e2, ocorridos em processo diferentes e estampilhados com um relógio lógico de Lamport, C(e1)=5 e C(e2)=8, respeita-se sempre a ordem causal se se entregar a mensagem correspondente a e1 antes da mensagem correspondente a e2 em todos os processos.

iv. Dados dois eventos, e1 e e2, com e1 a ocorrer no processo 1 e e2 no processo 2, é possível que os mesmos fossem estampilhados com as seguintes estampilhas vectoriais: C(e1)=[5 2 3] e C(e2)=[5 4 1]?

v. Dados dois eventos de envio de mensagem multi-ponto, e1 e e2, com e1 a ocorrer no processo 1 e e2 no processo 3, e sabendo que os mesmos foram estampilhados com as seguintes estampilhas vectoriais: C(e1)=[5 2 3] e C(e2)=[3 4 7], respeita-se sempre a ordem causal se se entregar a mensagem correspondente a e1 antes da mensagem correspondente a e2 em todos os processos.

5. Considere o seguinte esquema temporal representando a propagação de mensagens num sistema composto por três processos. As setas indicam a propagação de mensagens entre os vários processos. O início de uma seta indica o envio de uma mensagem. O fim de uma seta representa a recepção de uma mensagem num processo.



Suponha que pretende usar relógios vectoriais para atribuir estampilhas temporais aos vários eventos. Indique um valor possível da estampilha temporal para cada um dos seguintes eventos.

Envio de m1:

Envio de m2:

Envio de m3:

Envio de m4:

Recepção de m1 no processo 2:

6. Considere um sistema distribuído de gestão de ficheiros semelhante ao proposto no trabalho prático, composto por um servidor de nomes e um servidor de conteúdos de ficheiros. Neste sistema, a remoção de um ficheiro implica a remoção do conteúdo do ficheiro no servidor de conteúdos e a remoção da informação associada ao nome do ficheiro no servidor de nomes. Suponha que cada servidor pode decidir no momento em que recebe a operação se pode executar a operação ou não. Neste caso, seria possível garantir a atomicidade da execução das operações usando o protocolo *one-phase commit* ou seria necessário usar o protocolo *two-phase commit?* No primeiro caso, explique como. No segundo caso, explique porquê.

Basta usar o **one-phase commit** porque / É necessário usar o **two-phase commit** porque
(risque o que não interessar)

7. Suponha que pretende desenvolver um sistema de ficheiros distribuído para um ambiente de rede local em que o modo de acesso típico dos ficheiros é o seguinte:
   - Um ficheiro é aberto num de dois modos: leitura ou leitura/escrta.
   - Um ficheiro permanece aberto durante um período de tempo elevado (em qualquer modo);
   - O acesso ao ficheiro é "aleatório" no sentido em que são acedidas zonas do ficheiro não contíguas;
   - As escritas são efectuadas por sobreposição do conteúdo anterior;
   - Existem zonas do ficheiro que são lidas com grande frequência;
   - Existem zonas do ficheiro que são modificadas muito raramente (algumas das quais são acedidas com frequência elevada);
   - Os ficheiros são de grande dimensão.

   a. Nestas condições, e assumindo que não há falhas no sistema de comunicações, o sistema de gestão do cache implementado pelo sistema CIFS/SMB (baseado na utilização de op. locks) impede qualquer tipo de acesso a versões desactualizadas dum ficheiro? Justifique a resposta.

SIM, porque / NÃO, porque (risque o que não interessar)

   b. Indique que (se alguma) alterações introduziria ao mecanismo de gestão da cache usado pelo sistema NFS para se adaptar ao cenário apresentado (tenha em consideração o modo como o sistema adiciona e remove dados da cache). Justifique a resposta.
