FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Departamento de Informática

Mestrado Integrado em Engenharia Informática
Sistemas Distribuídos – 2º teste, 2 de Junho de 2016
2º Semestre, 2015/2016

NOTAS: Leia as questões atentamente antes de responder. **O teste é sem consulta. A duração do teste é 2h00min**. O teste contém **7** páginas.

Nome: _____ **Número**: _____

1) Indique se cada afirmação é **[V]erdadeira** ou **[F]alsa** (**nota: respostas incorretas descontam**):

___ Num pedido Ajax efetuado usando o suporte nativo dos browsers, a resposta é atendida assincronamente quando disponível.

___ O JSON é um formato de codificação de informação muito eficiente por usar uma representação binária dos dados.

___ Numa operação acedida usando REST, é possível sinalizar que se tentou aceder a um recurso não existente devolvendo o código HTTP "404 Not Found".

___ É impossível aceder a um serviço disponibilizado usando SOAP através duma conexão segura (https).

___ Diz-se que um algoritmo tem segurança futura perfeita se mesmo que no futuro forem comprometidas as chaves secretas dum servidor não é possível a um atacante decifrar uma comunicação efetuada por esse servidor no passado (e que tenha sido armazenada).

___ Um ataque por *replaying* consiste em um atacante guardar as mensagens trocadas por um cliente e um servidor e voltar a executar a comunicação repetindo as mensagens enviadas por um dos parceiros.

___ Um ataque de *Denial of Service* tipicamente permite a um atacante obter informação a que não devia ter acesso.

___ No protocolo SSL, o servidor envia para o cliente um certificado X.509 que contém a sua chave privada.

___ No sistema CIFS, um cliente que obtém um *oplock* partilhado (para leitura) tem de fazer cache do ficheiro completo.

___ Num sistema de caching distribuído com *delayed write*, uma escrita pode-se perder em caso de falhas.

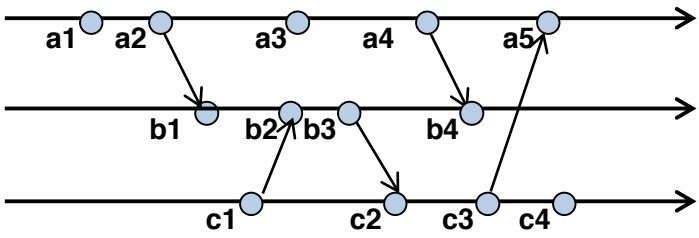
___ No sistema Coda, quando para um ficheiro, o servidor A tem o vetor versão [2 3] e o servidor B tem o vetor versão [2 7], para sincronizar os servidores deve-se copiar o ficheiro do servidor B para o servidor A.

___ No protocolo de replicação primário-secundário, o primário deve esperar pela confirmação (*acknowledge*) de todos os secundários antes de responder ao cliente.

___ No DNS é possível obter, para uma máquina, um IP desatualizado.

___ A síntese segura (*secure hash*) de um URL é um nome puro.

2) Considere um sistema distribuído com três processos, em que ocorrem os eventos assinalados a1, a2, ... As setas indicam o envio de uma mensagem.



a) Neste contexto, indique todos os eventos que aconteceram antes de:

a3: **c4:**

b) Suponha que pretende identificar os eventos com relógios lógicos de Lamport. Indique o valor para cada um dos seguintes eventos, sabendo que o primeiro evento de cada processo será identificado com o relógio 1 e que cada relógio será incrementado sempre pelo menor valor possível.

a2: _____ a3: _____ b2: _____ c2:_____a5: _____

c) Com base no valor dos relógios lógicos de Lamport de quaisquer dois eventos e1 e e2, é possível saber se e1 aconteceu antes de e2? Explique como ou porque não.

Sim, porque... / Não, porque...

Nome: _____ **Número**: _____

Considere o seguinte protocolo em que a Alice (A) envia uma mensagem M para o Bob (B). Neste protocolo também intervém um centro de distribuição de chaves (KDC), que é uma entidade confiável que armazena chaves simétricas partilhadas com os principais, i.e., a chave simétrica Ka é partilhada entre A e o KDC e a chave simétrica Kb é partilhada entre B e o KDC. H(M) representa o hash seguro de M.

(1) Alice -> Bob: A, {B, Ks, t}Ka, {M, Na}Ks, H(M)

(2) Bob -> KDC: A, {B, Ks, t}Ka

(3) KDC -> Bob: {A, Ks, Nb}Kb

(4) Bob -> Alice: H(M+Na)

a) Terá o Bob garantias que **M** provém da Alice? **Justifique**.

Sim, porque… / Não, porque…

b) O que prova à Alice que Bob conseguiu ler a mensagem que esta lhe enviou? **Justifique**.

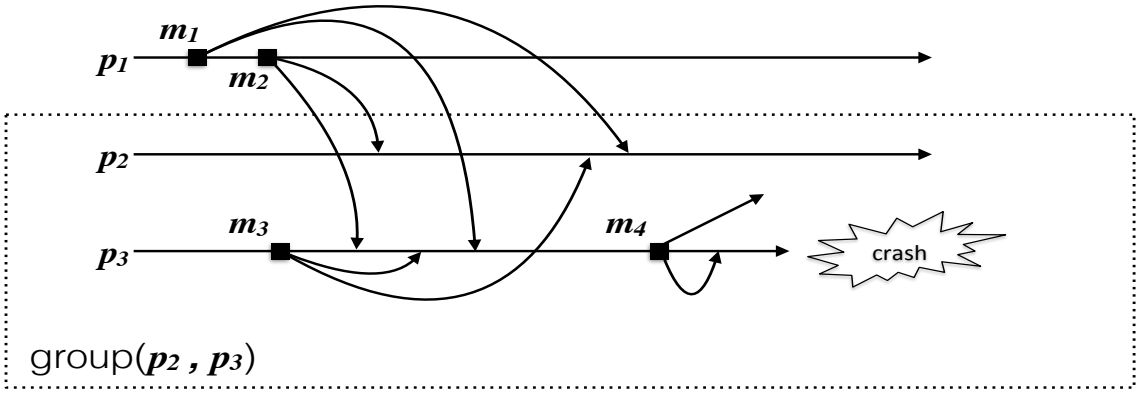
Sim, porque… / Não, porque…

c) Apresente um protocolo seguro que permita à Alice enviar uma longa mensagem confidencial ao Bob e receber em troca (do Bob) uma prova que este leu essa mensagem. Admita que o Bob possui um par de chaves de criptografia assimétrica, cuja chave pública é do conhecimento da Alice.

(1) Alice -> Bob:

(2) Bob -> Alice:

4) Considere o seguinte diagrama que ilustra um padrão de comunicação em grupo, envolvendo três processos, dos quais, o **p2** e **p3** pertencem ao grupo destinatário de todas as mensagens e **p1** que apenas participa como emissor. As setas indicam o momento em que ocorre a entrega da mensagem no processo.



a) Considerando o par de mensagens **(m1, m2)**, ambas enviadas por **p1**, assinale todas as ordens de entrega que são compatíveis com o apresentado no diagrama.

1. Sem ordem
2. FIFO
3. Total
4. Causal

b) Considerando o par de mensagens **(m1, m3)**, assinale todas as ordens de entrega que são compatíveis com o apresentado no diagrama.

1. Sem ordem
2. FIFO
3. Total
4. Causal

c) Considerando o par de mensagens **(m2, m3)**, assinale todas as ordens de entrega que são compatíveis com o apresentado no diagrama.

1. Sem ordem
2. FIFO
3. Total
4. Causal

d) Considere a mensagem **m4**, enviada por **p3** ao grupo pouco antes deste processo falhar de forma definitiva (*fail stop*). Caso **m4** não seja entregue a **p2** será possível afirmar que a primitiva de comunicação em grupo utilizada implementa aquilo que se designa por "reliable multicast". **Justifique**.

Nome: _____ **Número**: _____

5) Foi criado um sistema que permite aos seus utilizadores aceder a uma galeria partilhada de imagens e fotografias. Neste sistema, uma galeria consiste em múltiplos álbuns e cada álbum consiste num conjunto de imagens. Os utilizadores acedem à galeria através de um programa cliente que implementa a interface gráfica do sistema.

   O serviço disponibiliza as seguintes operações: (1) criar álbum, dado o nome do álbum; (2) remover álbum, dado o nome do álbum; (3) listar as imagens dum álbum, dado o nome do álbum; (4) adicionar uma imagem a um álbum, dado o nome do álbum e da imagem; (5) obter o conteúdo duma imagem, dado o nome do álbum e da imagem.

   a) Caso pretendesse disponibilizar o serviço usando REST, indique a operação e o URL que usaria para cada operação do serviço.

(1)

(2)

(3)

(4)

(5)

   b) Um estudo revelou que a popularidade dos álbuns não é igual entre os utilizadores. Cada utilizador apenas consulta ativamente um pequeno conjunto de álbuns do seu interesse. O sistema revelou-se muito popular e já contém centenas de álbuns que recebem novas imagens continuamente.
   Uma consequência da popularidade do sistema é que a experiência de utilização está a degradar-se devido à dificuldade crescente que os utilizadores têm para encontrar conteúdos do seu interesse. Pretende-se resolver o problema recorrendo a um *middleware* publish/subscribe (editor/assinante). Indique qual o tipo de sistema pub/sub que escolheria para permitir alertar os utilizadores que os álbuns do seu interesse têm novos conteúdos. Descreva um esboço da sua solução.

6) Considere um serviço de armazenamento de ficheiros que é materializado por um número (variável) de servidores. Cada servidor fornece três operações: ***guardar um ficheiro***, **apagar um ficheiro**, e **listar um conjunto de ficheiros**. O sistema usa replicação total num modelo *multi-master*, em que os clientes podem executar operações sobre qualquer servidor). Porém, os clientes apenas contatam um dos servidores (de entre os ativos) quando executam cada operação.

   O algoritmo de replicação é executado no cliente e o pseudo-código deste algoritmo encontra-se abaixo (Note que o algoritmo assume que os clientes têm acesso a uma lista de servidores ativos, denominada *serverList,* que é mantida atualizada por uma outra *thread em background*).

Algoritmo de replicação executado por cada cliente:

```
Every (x seconds) do:  //Infinite and periodic loop
  If( serverList.size() >= 2 ) then:
     (S1,S2) <- selectRandomPair(serverList)  // S1 != S2
     ListOfFilesInS1 <- S1.requestListOfFiles( ) //Remote operation
     ListOfFilesInS2 <- S2.requestListOfFiles( ) //Remote operation
     For Each ( File f in ListOfFilesInS1 ) do
        If ( f does not exist in ListOfFilesInS2 ) then:
           S2.requestSaveFile( f ) //Remote operation
        End If
     For Each ( File f in ListOfFilesInS2 ) do
        If ( f does not exist in ListOfFilesInS1 ) then:
           S1.requestSaveFile ( f ) //Remote operation
        End If
  End If
```

a) O algoritmo apresentado garante consistência eventual, considerando a definição que diz que: quando as operações de escrita (neste caso criação e remoção de ficheiros) terminarem num momento no futuro todas as réplicas, garantidamente, convergem para o mesmo estado? **Justifique** a sua resposta com base no estado final do sistema.

Sim/ Não , porque…

Nome: _____ **Número**: _____

b) O algoritmo apresentado poderá levar o sistema a ter um comportamento inesperado para os seus utilizadores? (considere aqui que um utilizador, quando realiza uma operação de escrita espera que os efeitos desta sejam permanentes e visíveis no sistema). **Justifique** a sua resposta. NOTA: as operações que alteram o estado do sistema são: criar ficheiro e apagar ficheiro.

Sim/ Não , porque...

c) Se considera que o algoritmo apresentado pode gerar comportamentos inesperados por parte do sistema, indique que mecanismo poderia ser usado para enriquecer o algoritmo de forma a evitar esses comportamentos e de forma sucinta como é que este poderia ser utilizado.