Número: Nome:

Sistemas Distribuídos

FCT NOVA
2021/22 - Teste 2
sem consulta ; duração total: 1h30
NOTA IMPORTANTE: o tamanho das caixas é indicativo da dimensão esperada da resposta e não um desafio para descobrir quem consegue fazer a letra mais pequena.

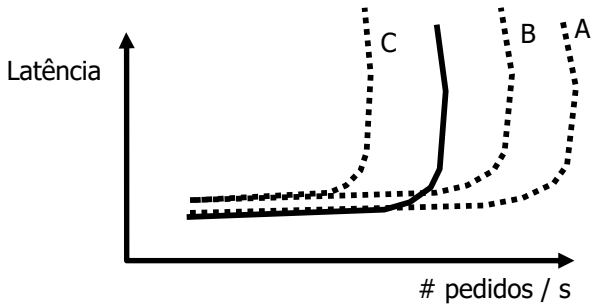
Número: _____ Nome: _____

As respostas erradas às perguntas V/F descontam até o equivalente ao valor da resposta certa correspondente. Para as perguntas de escolha de múltipla, o desconto é de $1/(n-1)$, com n o número de opções. A penalização acumula apenas no contexto da mesma pergunta. Em cada pergunta, a primeira resposta errada não desconta.

Questão 1
Para cada uma das seguintes afirmações, indique se é `[V]`erdadeira ou `[F]`alsa ou escolha a resposta apropriada, no caso das perguntas de escolha múltipla.

1. ____ Uma diferença importante do teste de sistemas distribuídos face ao teste de software que executa num só processo é que é necessário considerar as falhas de comunicação. (V/F)

2. Avaliou-se um sistema replicado com uma carga de trabalho em que há 50% de escritas e 50% de leituras. A linha contínua representa o desempenho do sistema quando existe apenas 1 servidor.
   ____ Qual a linha que representa uma configuração do sistema com 3 servidores, sendo que a informação é replicada em todos os servidores - uma operação de escrita apenas retorna ao cliente após ter escrito numa maioria de servidores, e as operações de leitura podem ser feitas num qualquer servidor? (A/B/C)
   ____ Qual a linha que representa uma configuração do sistema com 3 servidores, sendo que a informação é particionada pelos servidores e os clientes sabem que servidor devem contactar para executar uma operação? (A/B/C)
   Nota: Caso considere que duas linhas são possíveis, escolha a lexicograficamente menor ($A < B < C$).



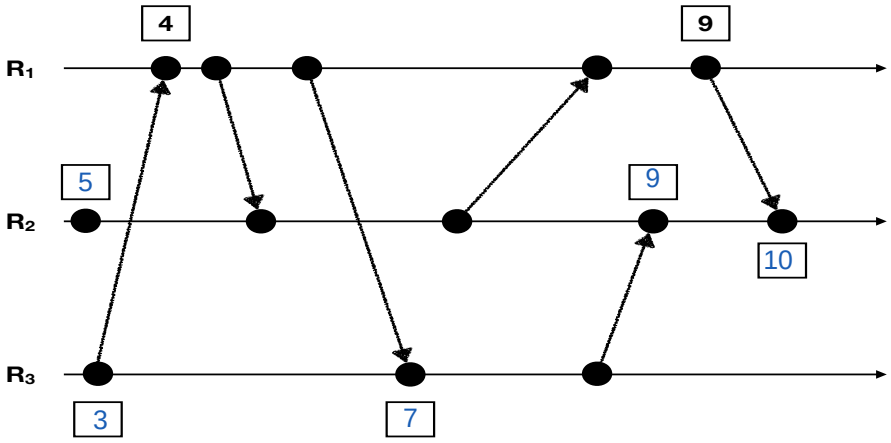
3. V ____ Se o evento e_1 aconteceu antes do evento e_2, então é porque o evento e_1 pode ter sido a causa de ocorrência do evento e_2. (V/F)
4. F ____ Num sistema distribuído, dados três eventos, e_1, e_2 e e_3, se e_1 aconteceu antes de e_2 e e_1 aconteceu antes de e_3, então e_2 aconteceu antes de e_3. (V/F)
5. F ____ Dados dois relógios de Lamport relativos aos eventos e_1 e e_2, é sempre possível saber se correspondem a eventos que estão causalmente relacionados (i.e., e_1 aconteceu antes de e_2 ou e_2 aconteceu antes de e_1) ou se são eventos concorrentes. (V/F)
6. V ____ Dadas duas histórias causais relativas aos eventos e_1 e e_2, é sempre possível saber se correspondem a eventos que estão causalmente relacionados (i.e., e_1 aconteceu antes de e_2 ou e_2 aconteceu antes de e_1) ou se são eventos concorrentes. (V/F)
7. V ____ Dados dois relógios vetoriais relativas aos eventos e_1 e e_2, é sempre possível saber se correspondem a eventos que estão causalmente relacionados (i.e., e_1 aconteceu antes de e_2 ou e_2 aconteceu antes de e_1) ou se são eventos concorrentes. (V/F)
8. V ____ No algoritmo de replicação primário/secundário, quando um secundário toma conhecimento que existe um novo primário deve deixar de aceitar operações vindas do primário antigo. (V/F)
9. V ____ O sistema Dynamo usa vetores versão para detetar a existência de escritas concorrentes. Seria possível obter a mesma funcionalidade substituindo os vetores versão por histórias causais, mas seria menos eficiente. (V/F)
10. V ____ O mecanismo de *delayed write* usado na gestão da cache no NFS permite melhorar o desempenho do sistema mas pode causar problemas de fiabilidade, em que os dados escritos pelos clientes se podem perder. (V/F)
11. V ____ No sistema de gestão de cache baseada em *op locks*, caso um cliente tenha um *op lock* partilhado, pode fazer cache dos ficheiros, mas apenas pode ler o conteúdo do ficheiro. (V/F)

Número: Nome:

Questão 2
Para cada uma das seguintes afirmações, indique se é [V]erdadeira ou [F]alsa ou escolha a resposta apropriada, no caso das perguntas de escolha múltipla.

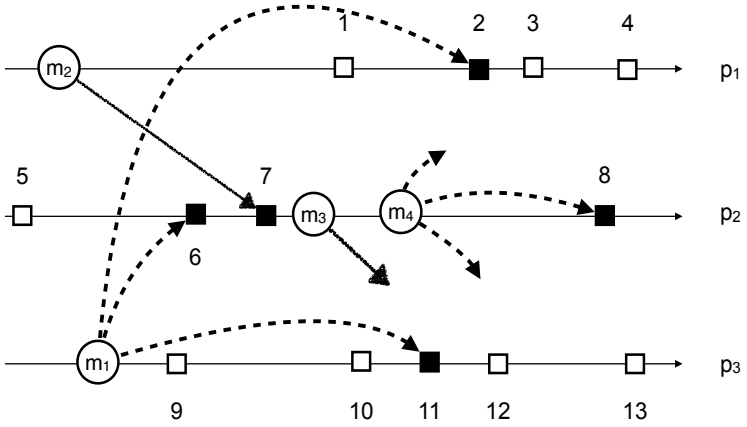
1. V Num sistema de comunicação em grupo, a filiação do grupo (elementos que pertencem ao grupo) pode variar ao longo do tempo.
2. V No sistema Kafka, ao criar um cliente, é possível indicar que se pretendem receber todas as mensagens passadas ou apenas as mensagens novas. (V/F)
3. V Um sistema de *message queues* pode ser usado para implementar um sistema de invocação remota assíncrono fiável. (V/F)
4. V Num sistema com dois processos, se as mensagens são entregues por ordem FIFO também são entregues por ordem causal. (V/F)
5. F Num sistema com N processos ($N > 2$), se as mensagens multicast são entregues por ordem FIFO também são entregues por ordem causal. (V/F)
6. V Um sistema para ser confiável deve ser altamente disponível, tolerante a falhas e seguro na presença de ataques. (V/F)
7. V Se se descobrir um método para fatorizar um número em números primos em tempo constante, toda a criptografia assimétrica deixa de ser utilizável. (V/F)
8. V O protocolo de Needham-Schroeder com chaves secretas resolve o problema da distribuição de chaves secretas para a comunicação segura entre duas entidades que estejam registadas no centro de distribuição de chaves. (V/F)
9. F Para que um utilizador se autentique perante um servidor que guarda pares *(nome de utilizador, chave secreta)* de todos os utilizadores é necessário que o utilizador envie a chave secreta para o servidor (i.e., não é possível criar um protocolo em que a chave secreta não passe entre o utilizador e o servidor).
10. V O algoritmo Diffie-Hellman permite criar um canal seguro entre dois parceiros de comunicação, sendo que todas as mensagens necessárias para o estabelecimento do canal seguro são passadas em claro.
11. ____ O protocolo TLS comprime os dados transmitidos pela rede antes de os cifrar.
12. F Ao aceder a um servidor HTTP, para um cliente verificar a validade dum certificado de chave pública recebido basta verificar que o mesmo se encontra assinado por um entidade de certificação na qual confia e que o certificado corresponde ao servidor HTTP a que o cliente se está a ligar.

Questão 3
Considere o seguinte diagrama temporal, correspondente a um sistema composto por três processos: R_1, R_2 e R_3, onde são representados os eventos relativos à comunicação entre processos que é ponto-a-ponto e uni-direcional. Os valores contidos, nas caixas junto aos eventos, correspondem a estampilhas de um relógio de Lamport. Preencha os valores em falta, utilizando para cada entrada o valor máximo admissível, tendo em conta que as estampilhas devem ser valores inteiros e o incremento é unitário.



Questão 4
Considere o seguinte diagrama que ilustra um padrão de comunicação, envolvendo 3 processos: p_1, p_2 e p_3. No total, são enviadas 4 mensagens: m_1 e m_4 pertencem a uma primitiva de comunicação em grupo, m_2 e m_3 são mensagens ponto-a-ponto. Para as mensagens m_1 e m_2 são indicados os respetivos eventos de recepção, assinalados com as setas a apontar para caixas pretas. Para as mensagens m_3 e m_4 o diagrama é omisso ou está incompleto. A comunicação é fiável e todos os processos deverão receber cada uma das mensagens. Responda às seguintes questões com base na informação presente no diagrama ou que se pode deduzir do mesmo.

Número: Nome:



a) Considerando apenas o par de mensagens (m_1, m_4), indique um padrão de entrega que respeite a ordem total. Havendo vários eventos possíveis, escolha os assinalados com o menor valor numérico.

$m_1 \rightarrow [2, 6, 11\ ]$	$m_4 \rightarrow [$ 3,8,12 $]$

b) Considerando apenas o par de mensagens (m_1, m_4), indique um padrão de entrega que viole a ordem total. Havendo vários eventos possíveis, escolha os assinalados com o menor valor numérico.

$m_1 \rightarrow [2, 6, 11\ ]$	$m_4 \rightarrow [$ 1,8,9 $]$

c) Considerando apenas o par de mensagens (m_1, m_4), indique um padrão de entrega que viole a ordem causal. Havendo vários eventos possíveis, escolha os assinalados com o menor valor numérico.

$m_1 \rightarrow [2, 6, 11\ ]$	$m_4 \rightarrow [$ 1,8,9 $]$

d) Considerando apenas o par de mensagens (m_1, m_3, m_4), indique um padrão de entrega que respeite a ordem total causal. Havendo vários eventos possíveis, escolha os assinalados com o menor valor numérico.

$m_1 \rightarrow [2, 6, 11\ ]$	$m_3 \rightarrow [$ 12 $]$	$m_4 \rightarrow [$ 3,8,13 $]$

e) Considerando apenas as mensagens (m_1, m_3, m_4), indique um padrão de entrega que viole a ordem causal. Havendo vários eventos possíveis, escolha os assinalados com o menor valor numérico.

$m_1 \rightarrow [2, 6, 11\ ]$	$m_3 \rightarrow [$ 10 $]$	$m_4 \rightarrow [$ 1,8,9 $]$

Para as seguintes afirmações, indique se são (V)erdadeiras ou (F)alsas.

f) V Os envios das mensagens m_1 e m_2 são eventos concorrentes.

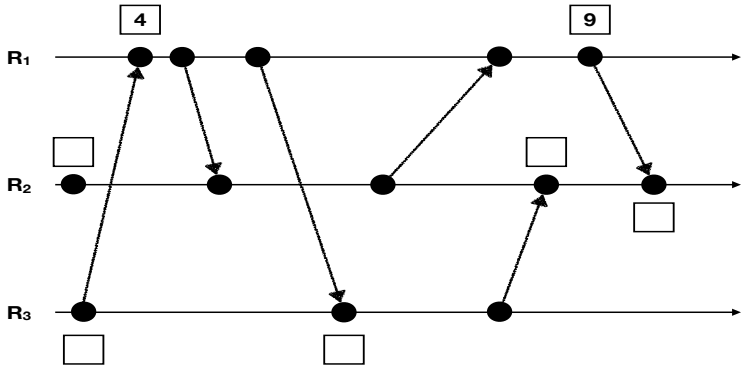
g) V Se os eventos de receção das mensagens m_1 e m_2 fossem trocados, as mensagens m_1 e m_2 seriam ainda concorrentes.

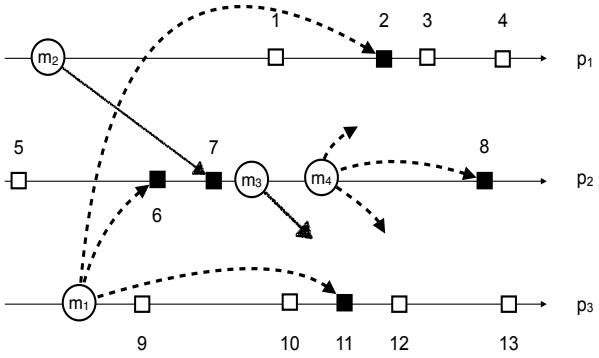
h) F O evento 5 não poderia ser o evento de receção da mensagem m_1, ou da mensagem m_2, por ser anterior ao envio de ambas.

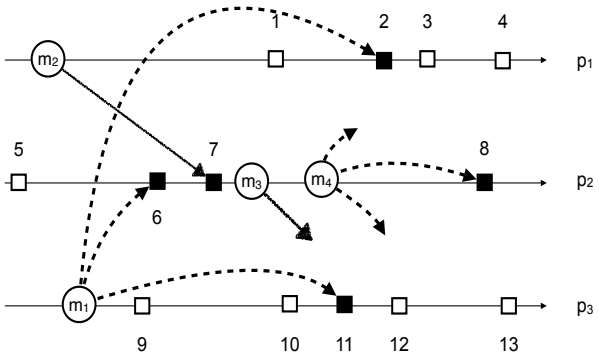
Número: Nome:

Rascunho

NOTA: esta página não será corrigida a menos que haja um referência explícita numa pergunta para uma resposta aqui.







Número: Nome:

Questão 5

Considere que pretende implementar o sistema de suporte a uma jogo de realidade aumentada, no qual os jogadores apenas podem observar e interagir com objetos virtuais quando se aproximam (fisicamente) do local em que esses objetos estão. O sistema mantém informação dos jogadores, incluindo a sua posição, pontuação e os objetos virtuais que o jogador apanhou, e informação sobre os objetos virtuais presentes no jogo, incluindo a sua posição e tipo.

a) Considere que a informação do sistema é mantido por um grupo de servidores, sendo que as operações que atualizam o estado do jogo são enviadas por multicast.

Indique **justificadamente** qual o tipo de ordem mais fraco (sem ordem/FIFO/causal/total) e fiabilidade (fiável/não fiável) que usaria na propagação de cada uma das seguintes operações.

Apanhar um objeto virtual (esta operação deve verificar que o objeto não foi apanhado por outro jogador):
Multicast fiável/não fiável e Ordem: sem ordem / FIFO / causal / total,

Multicast fiável para garantir ao jogador que este apanhou o objeto, com ordem total para que o sistema saiba quem foi o primeiro a apanhar o objeto sendo que apenas um jogador pode apanhar um dado objeto.

Atualização da pontuação do jogador (operação: adicionar X pontos):
Multicast fiável/não fiável e Ordem: sem ordem / FIFO / causal / total,

Multicast fiável para garantir que o jogador atualize os seus pontos, sem ordem porque não é relevante a ordem em que a adição é feita (se esta for a unica operação que altera os pontos)

b) Após apanhar um objeto (ou receber o objeto de outro jogador), um jogador pode transferir o objeto para um outro jogador. Para manter informação sobre estas transferências pretende-se manter uma sequência de registos, em que um jogador pode criar um registo a indicar que transfere um objeto que possui para outro jogador. O sistema do jogo também cria, sempre que um objeto é apanhado, um registo com essa informação.

Esta sequência de registos é mantida por um servidor, ao qual todos os jogadores podem aceder. Qualquer jogador deve poder, a partir da lista de registo mantida no servidor, computar que jogador possui que objeto.

Considere que cada jogador tem um par de chaves pública/privada (KpubA/KprivA para o jogador Alice), e que o sistema do jogo tem também um par de chaves pública/privada (KpubS/KprivS) e que os jogadores conhecem as chaves públicas de todos os outros jogadores.

Para registar a transferência do objeto O do jogador A para o jogador B, proponha um formato para o registo R_n a armazenar no servidor. A sua solução deve garantir a autenticidade e não repudiamento dos registos (de forma a que seja possível verificar se o jogador que solicita a transferência do objeto é o jogador que tem o objeto) e a sua integridade (de forma a que se o servidor for atacado e um utilizador tentar alterar um registo, essa tentativa de alteração é detetada).
NOTA: caso não saiba resolver o problema com todas as propriedades pretendidas, resolva para o subconjunto das propriedades que souber.

R_n : A, B, O,{H(A+B+O)}KprivA

O que garante a autenticidade e não repudiamento do registo?

A autenticidade e não repudiamento são garantidos pelo facto de apenas A ter a chave privada que foi utilizada para cifrar o hash seguro da mensagem.

O que garante a integridade dos registos?

A integridade também é garantida pela inclusão do hash cifrado por A, pois se alguem tentar alterar o conteúdo da mensagem, o hash decifrado não irá igualar ao hash da mensagem modificada

Questão 6
Considere que duas entidades partilham uma chave secreta K_s. Seria possível e útil estas entidades criar uma nova chave de sessão usando um protocolo baseado no protocolo Diffie-Helman? Se sim, explique porque é útil e indique que mensagens trocaria para criar a nova chave de sessão. Se não, explique porque não é possível ou útil.

Sim, porque... / Não, porque.... (risque o que não interessar)

Sim, porque permite que apenas as entidades envolvidas conheçam a chave de sessão, sem que esta tenha que ser partilhada por canais inseguros. Não requer a partilha prévia de segredos entre as duas entidades, e servem como segredos temporários obtidos no momento.

Questão 7
Considere que se pretende desenvolver um sistema de gestão de ficheiros composto por dois tipos de servidores: servidor de diretório, que mantém informação sobre os ficheiros que existem e os servidores de ficheiros em que estão armazenados; e servidores de ficheiros, que mantêm o conteúdo dos ficheiros. Os ficheiros são identificados por um identificador único (é responsabilidade dos utilizadores gerarem estes identificadores únicos).

Para replicar a informação nos servidores de diretório usa-se o Kafka. Por simplicidade, considere que existe apenas uma operação que altera o estado do servidor de diretórios `-writeFile`, que permite **criar** ou **modificar** um ficheiro. Considere o pseudo-código seguinte para implementar a operação writeFile.

```
function writeFile( fileid, contents)
    try
       filesrvs = selectFileServers( fileid)     // seleciona servidores para gravar ficheiro
       FOR srv IN filesrvs:                      // escreve conteúdo nos servidores de ficheiros
          filesrv.writeFile( fileid, contents)
                                                 // publica operação e espera pelo resultado
       version = kafka.publish( writeFileInfo( fileid, filesrvs))
       SyncPoint.waitForResult( version)
       return 204 No Content
    catch Exception                              // Em caso de erro de comunicação
       return 500 Server Error

Background thread
   forever
      op = kafka.receive()      // recebe uma operação
      res = op.exec()           // executa operação
      SyncPoint.registerResult( op.version, res)
```

a) O código anterior pode levar a que alguns servidores fiquem com "lixo", i.e., dados armazenados que não serão acedidos pelo utilizador. Justifique.

Sim, porque... / Não, porque.... (risque o que não interessar)

Sim, porque a operação de writeFile não garante que os ficheiros fiquem atualizados em todos os servidores, i.e. o writeFile pode falhar. Contudo, basta um write falhar para ficarmos com "lixo".

b) Caso as operações de escrita terminem sempre com sucesso e para um dado `fileid` sejam selecionados sempre os mesmos servidores, o código anterior garante que os servidores de ficheiros terão o mesmo estado quando param de haver operações de escrita? Justifique.

Sim, porque... / Não, porque.... (risque o que não interessar)

Sim, porque todas as operações de escrita terminam com sucesso, logo todos os servidores de ficheiro têm o mesmo estado.