Sistemas Distribuídos

Caderno de Problemas
v.2024.1

7 de fevereiro de 2024

Resumo

Este caderno de exercícios apresenta uma colectânea de perguntas que cobrem as matérias leccionadas na cadeira de Sistemas Distribuídos.

Versão 2024.1: Primeira versão da edição de 2024 da coletânea.

Problemas

RPC.

Questão 1 Indique uma vantagem e uma desvantagem de usar UDP como protocolo da camada de transporte de um sistema distribuído.

Questão 2 Qual destas não é uma característica do canal de comunicação de um socket do tipo stream?

- Canal com ligação.
- Canal bidirecional.
- Canal fiável.
- Interface tipo mensagem.

Questão 3 Um dado RPC, quando pretende invocar uma operação remota, simplesmente envia o pedido uma vez usando como base o protocolo UDP e aguarda até um tempo limite por resposta; caso a receba dentro do tempo limite, devolve o retorno ao cliente; caso contrário, devolve erro. Indique qual a semântica garantida por este RPC.

Questão 4 Assuma que um RPC oferece semântica pelo menos uma vez. Considere um sistema que usa este RPC para manter informação sobre o número de pessoas numa praia. Assuma que em dado momento o número de pessoas numa dada praia é de 1000, e é chamada a função remota `incrementar(500)` e o cliente RPC da aplicação recebe como resposta OK. A seguir, o cliente corre a função remota `obter_num_pessoas()`. Qual dos seguintes resultados não pode ser retornado?

- 1000
- 1500
- 2000
- 3000

Questão 5 Qual das seguintes funções não é idempotente?

- `set(object,value)`: atribui o valor `value` ao objeto `object`
- `update_log(string,position)`: atualiza um `log` (uma lista) com uma `string` na sua posição `position`.
- `reset(var)` limpa a variável `var` indicada
- `increment(var)`: incrementa o valor da variável `var`

Questão 6 Das seguintes tecnologias para comunicação remota, qual a menos eficiente?

- Uma variante de gRPC que use o formato JSON para serialização.
- O gRPC original.
- Web Services baseados em SOAP.
- Web Services baseados em REST.

Questão 7 Indique uma vantagem e uma desvantagem de Web Services baseados em SOAP, quando comparado com gRPC.

Questão 8 Qual destas não é uma característica do gRPC?

- Usa HTTP/2.
- Permite implementar servidores apenas nas linguagens Java e Go.
- Pode ser usado sobre HTTPS para garantir confidencialidade, autenticidade e integridade na comunicação.
- É um formato mais eficiente do que Web Services baseados em JSON.

Questão 9 Qual o serviço de nomes suportado pelo gRPC por omissão?

- DNS
- Zookeeper
- Protobuf
- O gRPC não suporta nenhum serviço de nomes.

Questão 10 Defina as mensagens e as funções remotas em protobuf para o RPC da Questão 4.

Sincronização de relógios físicos.

Questão 11 Usando o algoritmo de Cristian, um cliente tenta sincronizar o seu relógio com um servidor de tempo, sabendo que a rede que liga ambas as máquinas tem um tempo mínimo de propagação de 1 ms. O cliente fez 3 tentativas, observando como RTT (round-trip time): 8 ms, 10 ms, 4 ms. Qual a precisão alcançada com esta sessão de sincronização?

Questão 12 Considere que se usa o Algoritmo de Berkeley para sincronizar o relógio de um grupo de 3 computadores, A, B e C. O computador A ("mestre") pergunta aos restantes qual o valor do seu relógio, e recebe como resposta de B e C os valores 13:00:15 e 12:59:55, respetivamente (formato HH:MM:SS). O relógio de A marca 12:59:35. Qual o valor com que cada um dos relógios deve ser ajustado? (Para simplificar, considere que o RTT das mensagens trocadas é igual a zero.)

§

Considere um sistema com 2 processos. Cada processo faz, de forma independente, uma leitura do relógio do outro processo e ajusta o valor do seu relógio para tentar seguir o outro. Assuma que os processosfazem as leituras dos relógio aproximadamente ao mesmo tempo. Considere a seguinte execução:

reading	source p_s	target p_t	*pedidoEnviado* (p_s's clock)	*pedidoRecebido* (p_s's clock)	*(valor enviado por p_t)*
$R_{1.2}$	p_1	p_2	200	216	220
$R_{2.1}$	p_2	p_1	208	222	204

Considere que o erro introduzido pelo desvio dos relógios durante este processo é descartável. Assuma que não sabe qual o tempo mínimo para enviar uma mensagem na rede, pelo que considere que pode ser 0.

Questão 13 Para cada uma das leituras apresentadas na tabela, indique quais os ajustes que p_1 e p_2 aplicam ao seu relógio. Indique também qual o erro associado a cada leitura.

Questão 14 No pior caso, qual a diferença entre os valores dos relógios de p_1 e p_2 após ambas as leituras terminated?

Questão 15 Assuma agora um algoritmo diferente em que p_1é considerado um mestre e apenas p_2 muda o valor do seu relógio. Esta técnica oferece uma melhor ou uma pior precisão na sincronização dos relógios?

Relógios Lógicos.

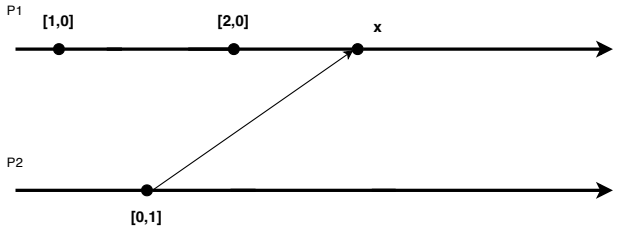
Questão 16 Considere dois eventos com os seguintes relógios vectoriais: A [1, 4, 7] e B [1, 8, 5]. Qual das seguintes frases é verdadeira?

- A aconteceu antes de B
- A aconteceu depois de B
- A é concorrente com B

- Esta situação nunca ocorre

§

Questão 17 Considere a sequência de eventos illustrada na figura seguinte, marcados com um relógio vectorial. Qual será o valor do relógio lógico vectorial do evento X?



Considere a execução distribuída apresentada na Figura 1. Todas as mensagens são enviadas ponto-a-ponto.

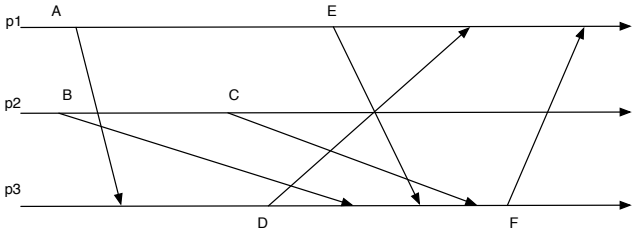


Figura 1: Logical and vector clocks

Questão 18 Assuma que se usa relógios vetoriais para estampilhar as mensagens e que os únicos eventos que são relevantes para receberem uma estampilha são os envios de mensagens. Assuma que, no início da execução ilustrada na figura, todos os relógios vetoriais de cada processo têm o valor seguinte: [100,100,100]. Indique os relógios vetoriais que são associados a cada mensagem enviada.

Questão 19 Assuma agora que se usam relógios de Lamport para estampilhar as mensagens enviadas. Assuma que, inicialmente, os relógios lógicos de todos os processos têm o valor 100. Indique os relógios lógicos das mensagens B e C.

§

Questão 20 Considere a execução distribuída apresentada na Figura 2. Todas as mensagens são enviadas ponto-a-ponto.

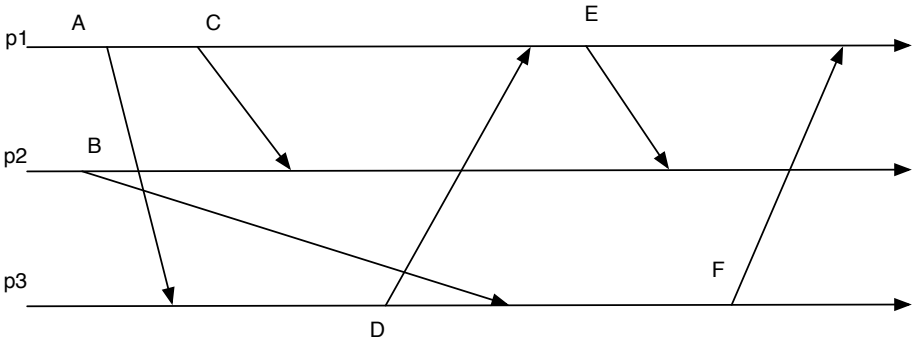


Figura 2: Relógios lógicos e vetoriais

Assuma que se usa relógios vetoriais para estampilhar as mensagens e que os únicos eventos que são relevantes para receberem uma estampilha são os envios de mensagens. Assuma que, no início da execução ilustrada na figura, todos os relógios vetoriais de cada processo têm o valor seguinte: [100,100,100]. Indique os relógios vetoriais que são associados a cada mensagem enviada.

Questão 21 Assuma agora que se usam relógios de Lamport para estampilhar as mensagens enviadas. Assuma que, inicialmente, os relógios lógicos de todos os processos têm o valor 100. Indique os relógios lógicos das mensagens B e C.

Questão 22 Considere as mensagens B e C: explique que informação sobre B e C pode ser inferida caso se use relógios vetoriais e que não pode ser inferida caso se use relógios de Lamport.

Exclusão mútua distribuída.

Considere o algoritmo distribuído de exclusão mútua de Ricart-Agrawala. Assuma um sistema com 3 processos, p_1, p_2 e p_3 e considere a execução ilustrada na Figura 3. Neste exemplo, inicialmente, nenhum processo está na secção crítica. O processo p_1 pede para entrar na secção crítica no tempo lógico 10. O processo p_3 também pede para entrar no tempo lógico 11.

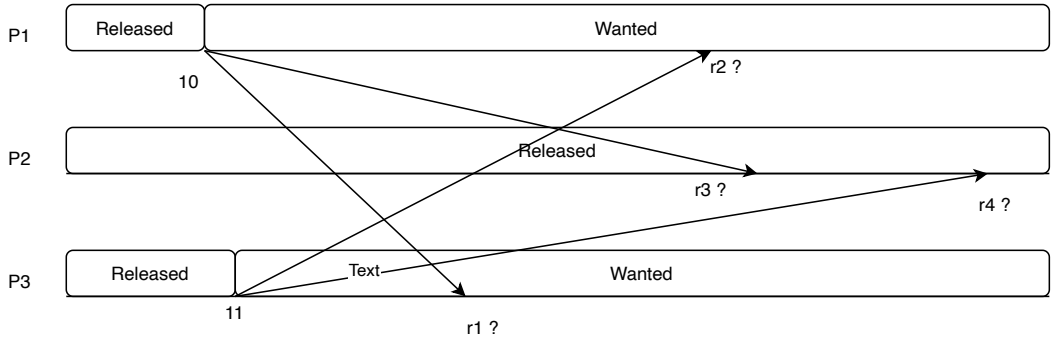


Figura 3: Execução do algoritmo Ricart-Agrawala

Questão 23 Para cada evento onde uma mensagem é recebida (i.e, r_1, r_2, r_3, e r_4), indique se o pedido é colocado em espera na fila ou se uma resposta positiva é retornada ao emissor do pedido.

Questão 24 Considere que o pedido do processo p_3 afinal levava o relógio lógico 10 (em vez de 11). O que mudaria?

§

Considere o algoritmo descentralizado de Maekawa para exclusão mútua. Considere um sistema com 3 processos, p_1, p_2 e p_3. Assuma os seguintes conjuntos de votos associados a cada processo: $V_1 = \{p_1, p_2\}$, $V_2 = \{p_2, p_3\}$, $V_3 = \{p_3, p_1\}$.

Neste algoritmo o estado de cada processo é capturado por 3 variáveis, nomeadamente:

- *state*, que pode ter os valores RELEASED, WANTED, ou HELD.
- *voted*, de tipo booleano.
- *pending*, uma fila de pedidos.

Considere que, num dado instante, o estado do sistema é o seguinte:

processo	p_1	p_2	p_3
state	HELD	RELEASED	RELEASED
voted	TRUE	TRUE	FALSE
pending	$\emptyset$	$\emptyset$	$\emptyset$

Questão 25 Assuma que, a partir deste estado, o processo p_2 tenta entrar na secção crítica (mas sem que o p_1 a liberte). Descreva o estado do sistema após todas as mensagens do algoritmo terem sido trocadas.

Estados globais.

Considere três processos, p_1, p_2 e p_3, envovlidos numa computação distribuída. Assuma que cada processo mantém um estado local que indica a quantidade de *tokens* que esse processo detém. Inicialmente, cada processo tem 100 tokens. Cada mensagem transfere 10 tokens de um processo para outro. Uma possível execução está ilustrada na Figura 4.

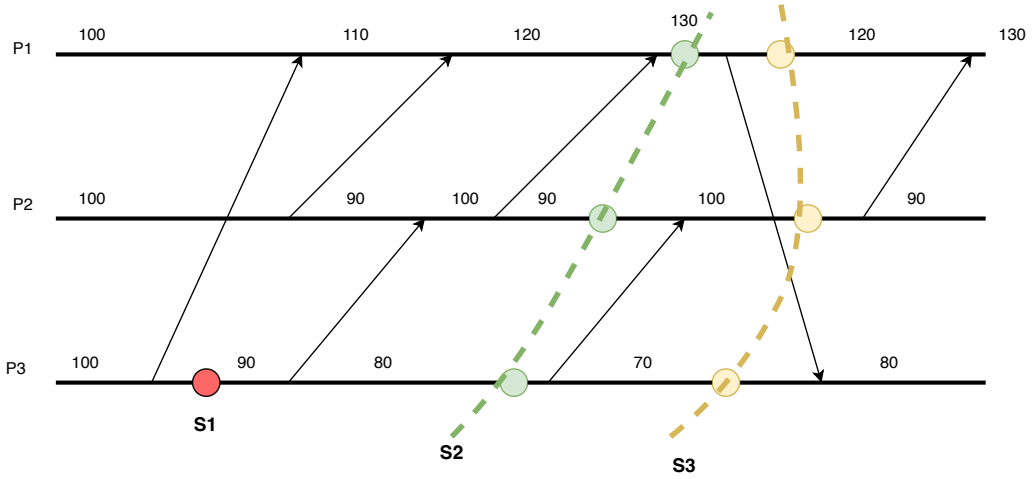


Figura 4: Execução distribuída

§

Considere os dois estados globais, $S2$ and $S3$, apresentados na figura.

Questão 26 O estado global $S2$ é coerente? Justifique.

Questão 27 O estado global $S3$ é coerente? Justifique.

§

Considere três processos, p_1, p_2 e p_3, envolvidos numa computação distribuída. Assuma que cada processo mantém um estado local que consiste em duas variáveis: o número de mensagens enviadas e o número de mensagens recebidas. Inicialmente, estas variáveis têm valor 0. Uma possível execução está ilustrada na Figura 5.

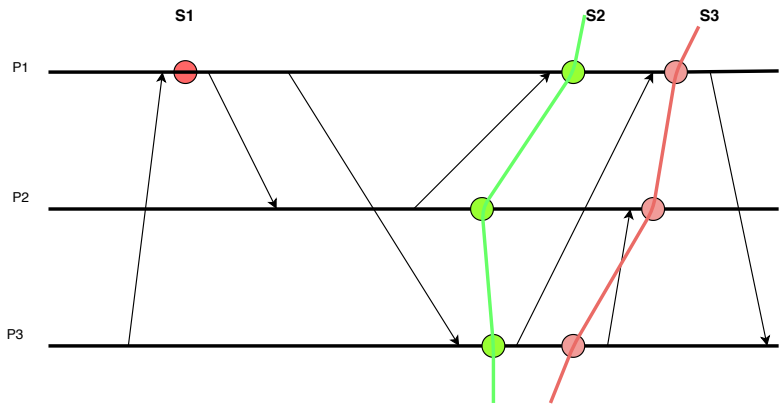


Figura 5: Execução distribuída

Considere os dois estados globais, $S2$ e $S3$, apresentados na Figura 5.

Questão 28 O estado global $S2$ é coerente? Justifique.

Questão 29 O estado global $S3$ é coerente? Justifique.

§

Considere quatro processos, p_1, p_2, p_3, e p_4, envolvidos numa computação distribuída. Uma execução está

ilustrada na Figura 6.

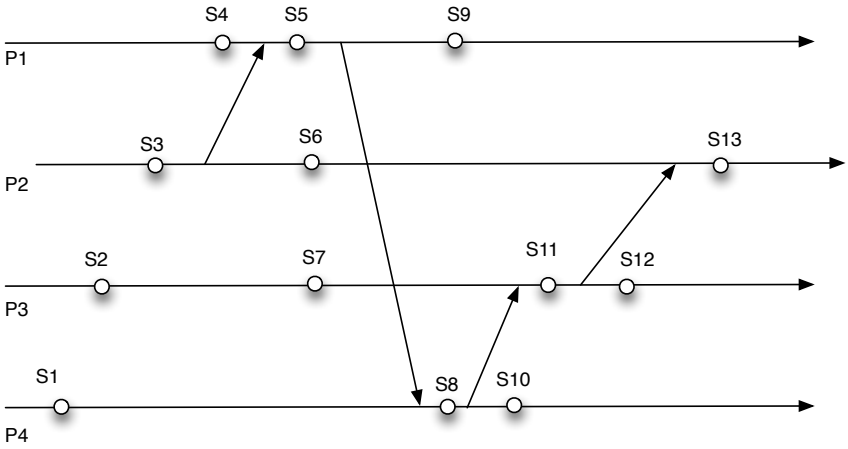


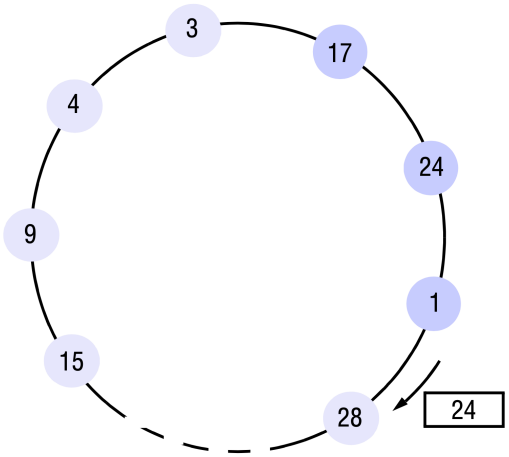
Figura 6: Execução distribuída

Questão 30 O estado $[S5, S6, S7, S8]$ é coerente? Justifique.

Questão 31 O estado $[S9, S6, S12, S10]$ é coerente? Justifique.

Eleição de líder.

Questão 32 Considere o algoritmo de eleição de líder num anel descrito no livro Coulouris et al.. Considere que o anel tem apenas os nós que aparecem na seguinte figura (assuma que os nós não apresentados na figura têm todos um identificador a 24).



Complete a frase: O nó 1, depois de encaminhar "election=24"...

- ...irá enviar a sua própria mensagem "election=1"
- ...ainda pode encaminhar mais 6 mensagens de "election".
- ...só irá encaminhar mais uma mensagem de "election"
- ...já não necessita de encaminhar mais nenhuma mensagem de "election".

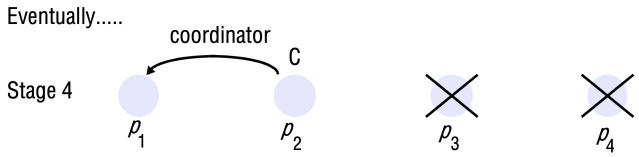
§

Questão 33 Comparando os algoritmos baseado em anel e "Bully" para o problema da eleição de líder, escolha a afirmação correta:

- No melhor cenário, o algoritmo de "Bully" consegue terminar a eleição trocando menos mensagens que o de anel.
- Ambos pressupõem que todos os processos conhecem previamente o identificador de todos os outros processos.

- Caso haja dois processos que concorrentemente decidem iniciar uma eleição, nenhum dos algoritmos assegura que todos os processos elegem o mesmo líder.
- O algoritmo baseado em anel exige um sistema síncrono, enquanto que o de "Bully" funciona em sistema assíncrono.

Questão 34 Considere o algoritmo de eleição "bully". Considere o cenário em que o nó C de declara coordenador.



Antes disso, o nó C:

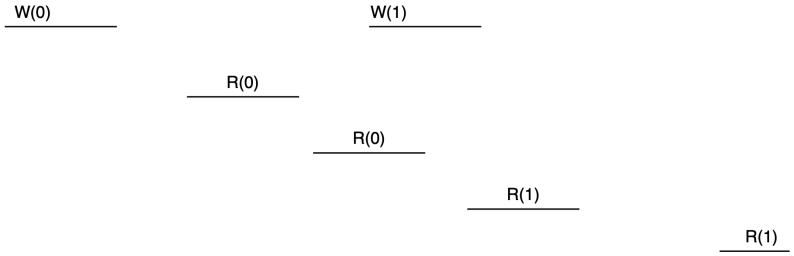
- Enviou uma mensagem de "election"só para p_3
- Enviou uma mensagem de "election"só para p_1
- Enviou uma mensagem de "election"só para p_3 e p_4
- Enviou uma mensagem de "election"para p_1, p_3 e p_4

Registos.

Questão 35 Condirere as execuções ilustradas na Figura 7
Indique qual corresponde a um registo *unsafe*, *safe*, *regular*, e *atómico*.

Questão 36 (*2 valores*) Considere as execuções de acesso a um registo ilustradas na Figura 8:
Apenas uma das execuções é válida para registos atómicos, outra só válida para registos regulares (mas não atómicos) e outra não é válida nem em registos atómicos nem em registos regulares. Diga, justificando, a que categoria corresponde cada execução.

Questão 37 Considere o seguinte ilustração que captura a execução de um registo distribuído. Este execução corresponde a um registo (indique a semântica mais forte que esta execução suporta):



- Unsafe
- Safe
- Regular
- Atomic

Questão 38 Considere o algoritmo ABD para concretizar registos atómicos distribuídos. Neste algoritmo, para realizar uma leitura...

- É preciso ler de uma maioria, escolher o valor mais recente, e escrever esse valor numa maioria
- É preciso ler de uma maioria, escolher o valor mais recente, e escrever esse valor na réplica local
- Basta ler de uma maioria e escolher o valor mais recente
- Basta ler a réplica local

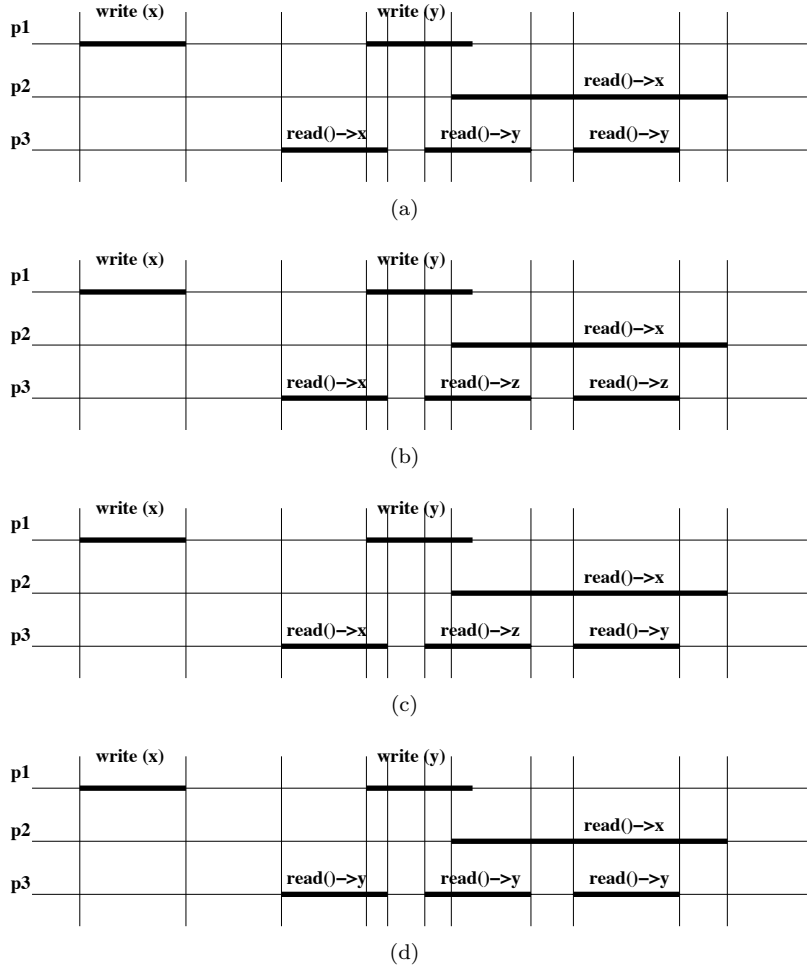


Figura 7: Distributed execution

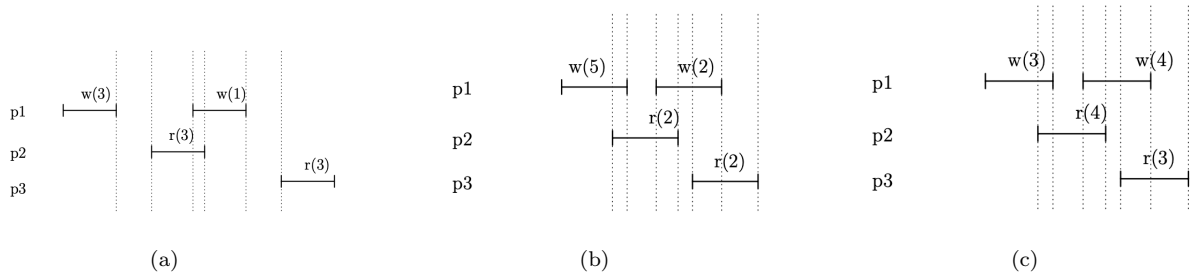


Figura 8: Execuções de registos

Questão 39 Considere um sistema de 3 servidores que oferecem um registo replicado através do algoritmo ABD, usando quóruns de maioria (cada servidor com o mesmo peso de voto). Considere que, num dado momento, há uma escrita a decorrer e o estado das suas réplicas é:

- Réplica A: valor=32, tag=5
- Réplica B: valor=10, tag=6
- Réplica C: valor=10, tag=6

Neste instante, um cliente envia um pedido de leitura. Que valor ele retornará à aplicação?

Questão 40 Considere um sistema de 3 servidores que oferecem um registo replicado através do algoritmo ABD, usando quóruns de maioria (cada servidor com o mesmo peso de voto). Considere que, num dado momento, há uma escrita a decorrer e o estado das suas réplicas é:

- Réplica A: valor=10, tag=32

- Réplica B: valor=12, tag=31
- Réplica C: valor=12, tag=31

Neste instante, um cliente envia um pedido de leitura. Que valor ele retornará à aplicação?

Espaços de tuplos.

Questão 41 Considere a operação Take num espaços de tuplos. Qual das seguintes frases é verdadeira:

- Esta operação não precisa de consenso
- Esta operação precisa de ser executada com ordem total em relação às outras operações de Take
- Esta operação é equivalente ao "Write"em registos
- Esta operação é equivalente fazer um "Read"seguido de um "Put"

Questão 42 Na abstração do espaço de tuplos, considere que, num instante inicial, existe um tuplo ("SD", "projeto", "grupo T90", "17v"). Um cliente pretende alterar essa informação, mudando o último atributo para "18v". Como pode fazer isso?

- Chama a primitiva "take"para eliminar este tuplo e depois chama a primitiva "put"para colocar o novo valor do tuplo.
- Usando a primitiva "update".
- Obtém um token de exclusão mútua, e de seguida chama "update"sobre o tuplo.
- Não é possível, pois os espaços de tuplos só permitem leituras.

Replicação fracamente coerente.

Considere o sistema replicado conhecido como a *Gossip Architecture*, onde as operações submetidas pelos clientes são primeiro adicionadas a um *log* (do servidor que recebe cada operação) e só depois disseminadas entre as restantes réplicas. Assuma um sistema com 3 réplicas. Considere que o estado de cada servidor é capturado pelos seguintes relógios vetoriais *valueTS*: S_1 =(2, 4, 2), S_2 = (1, 6, 7) e S_3 = (2, 5, 8). Considere também que o *replicaTS* de cada servidor é o mesmo (que o respetivo *valueTS*). Considere o caso em que o *front-end* do cliente C1 tem o seguinte *prev*=(2, 5, 5).

Questão 43 Assuma que C1 precisa executar uma leitura (*query*). Neste caso, indique que servidores são capazes de responder imediatamente ao pedido do cliente. Se um servidor puder responder, indique também o novo valor do relógio vetorial local ao cliente (atualizado após receber a resposta).

Questão 44 Assuma o mesmo estado inicial tal como acima para o cliente e para os servidores, mas agora considerando que o cliente pretende executar uma operação de escrita (*update*). Neste caso, indique que servidores podem aceitar o pedido e responder imediatamente ao cliente. Se um servidor puder responder, indique o relógio atualizado que o cliente passará a ter depois de receber a resposta desse servidor.

§

Considere o sistema replicado conhecido como *Gossip Architecture*. Assuma um sistema com 3 replicas, p_1, p_2 e p_3. Assuma que o ValueTS da réplica p_1 é (1,2,5). Assuma que a réplica p_1 tem as seguintes operações de escrita no seu *log*:

Client	OperationID	Source replica	Timestamp	prevTS
c_1	A	p_3	(1,1,5)	(1,1,0)
c_2	B	p_2	(1,2,5)	(0,1,0)
c_3	C	p_1	(2,3,2)	(2,1,0)

Questão 45 Que operações no log de p_1 já foram executadas localmente (i.e., estão estáveis) e quais não o foram ainda?

§

Questão 46 Consider algoritmo da Gossip Architecture lecionado nas aulas. Considere um cliente C com o relógio [3, 1, 7]. Este cliente tenta ler da réplica R1, cujo relógio é [3, 2, 6]. O que acontece?

- A réplica não responde e não altera o seu relógio
- A réplica não responde e muda o seu relógio para [3, 1, 7]
- A réplica responde e o relógio do cliente não é alterado
- A réplica responde e o relógio do cliente muda para [3, 2, 6]
- Não sabe/não responde

§

Replicação.

Questão 47 Considere que para tolerar faltas guarda o estado da aplicação periodicamente, de forma a poder relançar a mesma sem ter de começar do início. Esta técnica designa-se por:

- Replicação primário-secundário
- Replicação de máquina de estados distribuía
- Replicação optimista
- Salvaguarda-recuperação

Questão 48 Considere um sistema tolerante a faltas baseado em máquina de estados replicada. Neste sistema:

- Existe um primário que executa os pedidos e envia o resultados às restantes réplicas
- Não é preciso garantir que todas as réplicas recebem todas os pedidos. Basta que as mensagens sejam recebidas por uma maioria.
- Basta que os pedidos sejam entregues por ordem FIFO. Não é preciso ordem total.
- Os pedidos necessitam de ser entregues por ordem total.

Questão 49 Relativamente à replicação passiva (primário-secundário), qual das seguintes afirmações descreve corretamente o que primário faz após executar uma operação de escrita?

- Propaga o seu novo estado aos secundários e espera que todos (exceto os secundários em falta) respondam com ack, para finalmente responder ao cliente.
- Responde imediatamente ao cliente. Depois, em diferido (background), propaga o novo estado aos secundários.
- Guarda o novo estado em disco e responde ao cliente.
- Responde ao cliente, sem comunicar com as outras réplicas. As outras réplicas também receberam o update diretamente do cliente, logo também farão o mesmo.

Questão 50 Relativamente à replicação ativa e replicação passiva, qual das seguintes afirmações é correta?

- Na replicação ativa as réplicas executam as operações de escrita (updates) depois de chegarem a acordo sobre a ordem dos updates.
- Na replicação passiva, um cliente pode enviar uma operação de escrita (update) a qualquer réplica, que o aceitará.
- Em termos do teorema CAP, a replicação passiva só assegura C+A, enquanto que a replicação ativa só assegura A+P.
- Na replicação ativa a propagação de escritas (updates) consiste no envio aos secundários/backups do estado que foi modificado.

Difusão em ordem total.

Considero algoritmo de difusão totalmente ordenada inventado por Dale Skeen e usado no sistema ISIS original, que se baseia num acordo coletivo. Assuma que este algoritmo é usado para ordenar totalmente mensagens enviadas num grupo de 4 processos. Neste exemplo, os emissores são clientes que não pertencem ao grupo.

A Figura 9 ilustra o estado da fila de mensagens de cada processo num dado instante. Cada entrada na fila é um tuplo com o seguinte formato:

⟨ message_id, sender_id, sequence_number, state (**T**entative or **F**inal) ⟩.

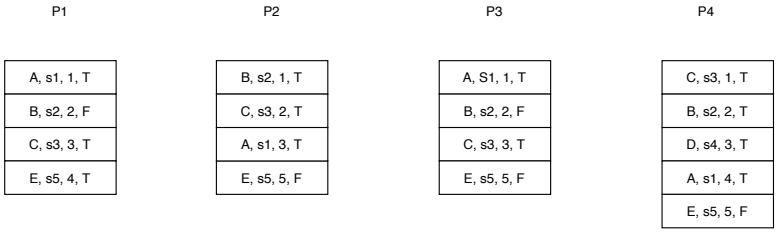


Figura 9: Execução em ordem total no ISIS

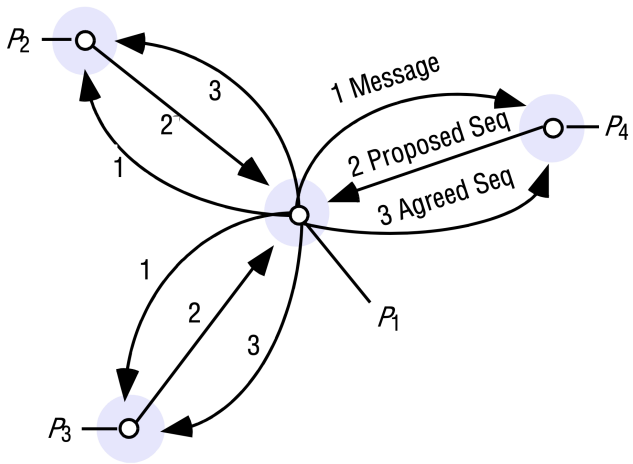
Questão 51 Nesta execução, qual será o número de sequência final atribuído à mensagem A?

Questão 52 Quando o processo P_1 passar a conhecer o número de sequência final da mensagem A, P_1 entregará alguma mensagem à aplicação? Se sim, qual?

Questão 53 (*0.5 point*) Assuma que, no estado ilustrado na figura, o processo P_2 recebe uma nova mensagem, F. Qual será o número de sequência que P_2 atribuirá a F?

§

Questão 54 Considere o algoritmo de acordo colectivo para estabelecer uma ordem total.

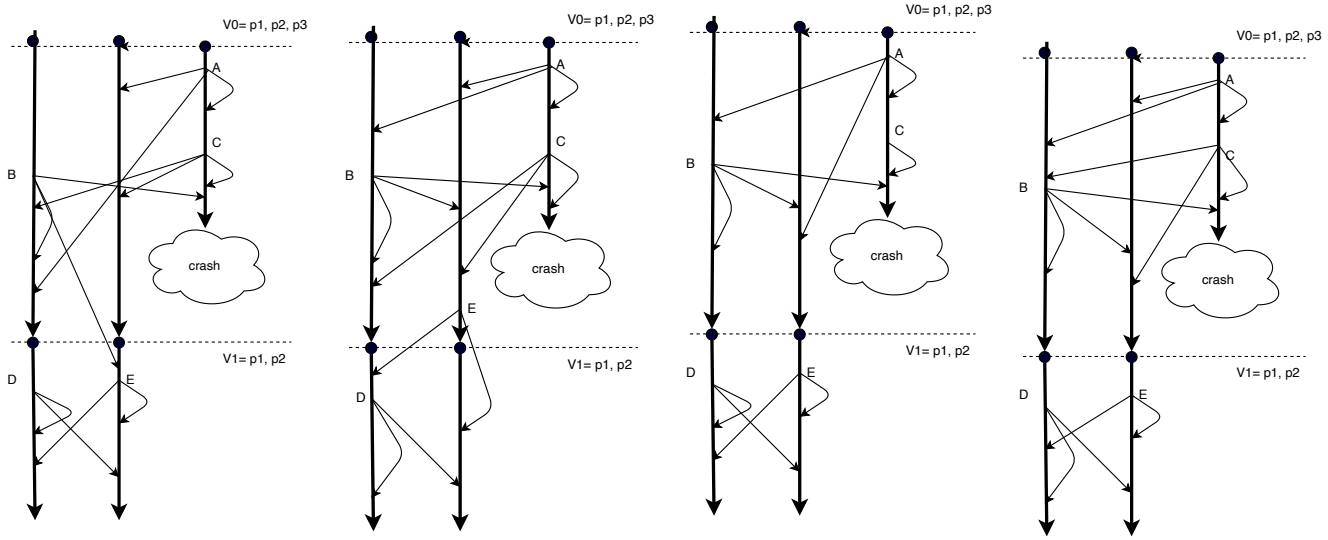


Neste algoritmo, o número de sequência final é:

- O mínimo dos números propostos pelos participantes
- Um número aleatório escolhido pelo emissor
- O máximo dos números propostos pelos participantes
- O valor do relógio fisico do emissor

Sincronia na Vista.

Considere as execuções representadas na Figura 10, que ilustram a comunicação entre três processos (p_1, p_2, and p_3) através de uma primitiva de difusão atómica. As letras representam a transmissão de mensagens,



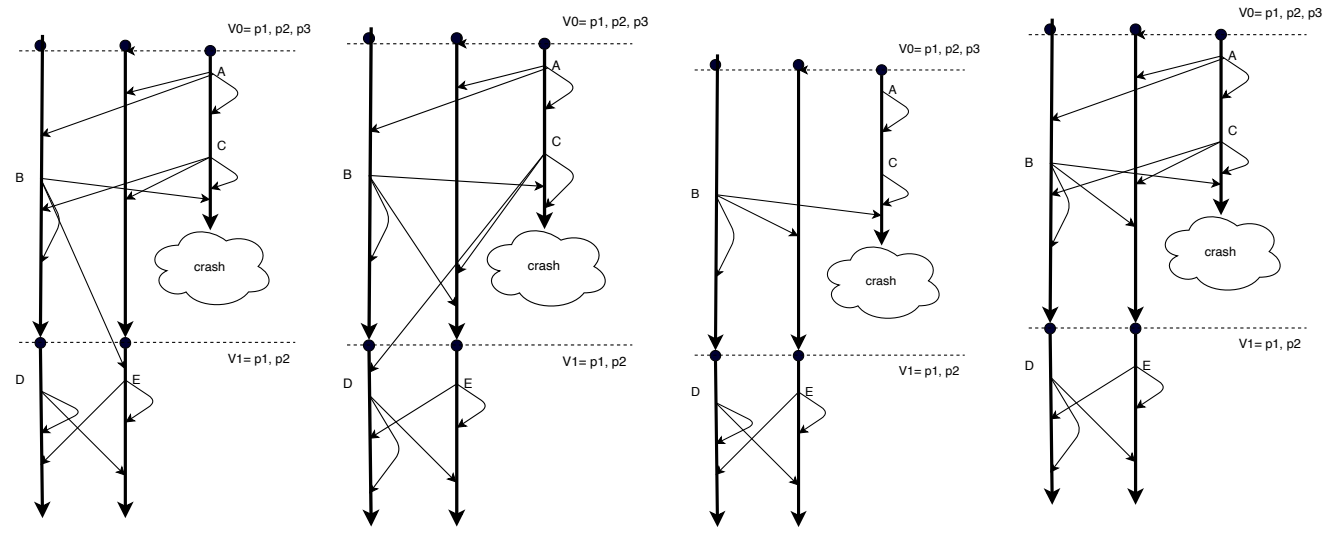
(a) Execução 1. (b) Execução 2. (c) Execução 3. (d) Execução 4.

Figura 10: Comunicação em Grupo

Questão 55 Quais são as excucações que violam a sincronia na vista? Justifique.

§

Considere as execuções representadas na Figura 11, que ilustram a comunicação entre três processos (p_1, p_2, and p_3) através de uma primitiva de difusão atómica. As letras representam a transmissão de mensagens,



(a) Execução 1. (b) Execução 2. (c) Execução 3. (d) Execução 4.

Figura 11: Comunicação em Grupo

Questão 56 Quais são as excucações que violam a difusão uniforme?

Questão 57 Quais são as excucações que violam a sincronia na vista? Justifique.

§

Considere um sistema que oferce sincronia na vista, difusão uniforme e respeita a ordem FIFO. Usando estas primitivas concretiza-se um sistema de replicaąã primário-secundário da seguinte forma:

when new_view (view)
cleanup ();
primary := lowest_id (view)

when receive request R from client
requests_pending := requests_pending $\oplus$ R

when I am the primary and requests_pending$\neq \emptyset$
R = remove_first (requests_pending)
$\langle$state-update, response$\rangle$ = execute (R);
vs-uniform-send (R, $\langle$state-update, response$\rangle$) to all members of the view

when vs-uniform-delivery (R, $\langle$state-update, response$\rangle$)
remove_from_list (R, requests_pending)
apply_update (state-update);
send_reply_to_client (response);

Considere que sempre que um nó faha é instalada uma nova vista de forma autom´atica.

Questão 58 Indique o código que deve ser executado pela função CLEANUP.

§

Consenso.

Questão 59 Considere o problema do consenso. Cada processo propõe um valor, e todos devem acordar num único output que é:

- Qualquer um dos valores propostos
- O valor proposto pela maioria
- Um vector com todos os valores propostos
- A soma de todos os valores propostos

Questão 60 Num sistema distribuído com 3 processos, executou-se o algoritmo "floodset consensus"(estudado nas aulas teóricas), em 3 rondas e usando a função "mínimo"para a escolha do valor. Assuma modelo síncrono.

Os processos propuseram os seguintes valores: p1 propôs 3, p2 propôs 10, p3 propôs 20. No entanto, a meio da primeira ronda, p1 falhou; consequentemente, a sua mensagem chegou a p2 mas não a p3. Os restantes processos mantiveram-se corretos até ao final do algoritmo.

Qual foi o valor acordado por p2 e p3?

Questão 61 Considere um serviço que oferece as seguintes garantias:

Module:

Name: ConsensoNaMaiorProposta (cnmp).

Events: v

Pedido: $\langle$ *cnmpPropor*, valor $\rangle$: Usado para propor um valor.

Resposta: $\langle$ *cnmpDecide*, resultado $\rangle$: Retorna o valor acordado

Proriedadess:

CNMP1: *Acordo Uniforme:* Dois processos nunca retornam valores distintos

CNMP2: *Integridade:* Um processo apenas retorna uma vez.

CNMP3: *Validade:* Se um processo retorna um valor v esse valor foi proposto por algum processo.

CNMP4: *MaiorValor:* Se um processo p retorna um valor v esse valor é igual ou maior que o valor proposto por p.

CNMP5: *Terminação:* Todos os processos correctem retornam mais cedo ou mais tarde.

Considere que pretende copncretizar este serviço num sistema sujeito a falhas por paragem, recorrendo a um detector de falhas perfeito, um serviço de difusão fiável uniforme, e um serviço de consenso. Complete o código abaixo definido a *condicao*, o *parametro*, e a *instrucao* em falta.

quando init()
processosactivos := $\mathcal{P}$
propostas := $\emptyset$
participantes := $\emptyset$
emconsenso := FALSE

quando cnmpPropor (v)
difusaoFiavelUniformeEnvia(v)

quando falha(p)
processosactivos := processosactivos $\setminus$ p

quando difusaoFiavelUniformeEntrega(v) de p
propostas := propostas $\cup$ v
participantes := participantes $\cup$ p

quando *condicao*
if emconsenso = FALSE
emconsenso := TRUE
consensoPropoe (*parametro*)

quando consensoDecide(v)
instrucao

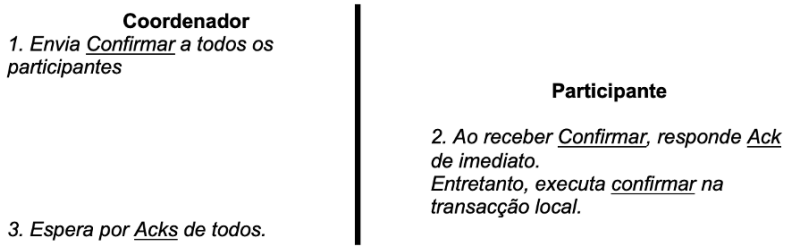
§

Trasações distribuídas.

Questão 62 Considere o protocolo conhecido por "2-phase commit"

- Este protocolo resolve o consenso de forma tolerante a faltas
- Este protocolo pode bloquear se um participante falhar
- Este protocolo pode bloquear se o coordenador falhar
- Este protocolo também é conhecido por "2-phase locking"

Questão 63 Considere o seguinte protocolo de confirmacção:



O protocolo de confirmação apresentado acima assegura a atomicidade da transacção distribuída? Justifique, ilustrando com um exemplo.

Questão 64 Considere o protocolo 2PC. Adicionalmente, considere que um dos participantes respondeu SIM e outro não responde. O coordenador deve:

- Decidir que o desfecho da transacção distribuída é ABORT.
- Reenviar o pedido para a transação que não responde tantas vezes quanto for preciso até conseguir resposta.
- Decidir que o desfecho da transacção distribuída é COMMIT.
- Contactar um terceiro participante para desempatar.

Questão 65 Considere o protocolo 2PC. Adicionalmente, considere agora que todas as transacções locais já responderam SIM ao Coordenador e estão à espera. Após um timeout sem que tenha chegado mensagem do Coordenador, as transacções:

- Têm obrigatoriamente de continuar à espera.
- Devem contactar o Coordenador e mudar o seu voto para NÃO.
- Podem decidir que a transacção já terminou e foi confirmada.
- Podem decidir que a transacção já terminou e foi cancelada.

Questão 66 Uma App Móvel pretende oferecer garantias transacionais aos seus utilizadores para reservas de pacotes de férias. Estes pacotes incluem: viagem de avião (flight), reserva de hotel (hotel) e aluguer de viatura (car). A App comunica com um servidor aplicacional, que contacta três serviços remotos com SOAP/HTTP. O sistema utiliza uma concretização do protocolo 2PC. Cada serviço oferece localmente as propriedades ACID e está a ser utilizado controlo de concorrência pessimista (strict 2-phase locking). O excerto abaixo mostra o código que 'e executado no servidor aplicacional quando o cliente da aplicação móvel pede para efectivar a compra do pacote de férias ("book" significa "reservar"):

```
1 void buyTravelPackage(int id) {
2     Tx t = openTransaction();
3     TravelPackage p = getTravelPackage(id);
4     p.flight.book(t);
5     p.hotel.book(t);
6     p.car.book(t);
7     closeTransaction(t);
8 }
```

Indique os números das linhas de código em que:

- É atribuído um identificador único à transacção
- O serviço de voos se junta à transacção
- Se inicia a votação do protocolo 2PC
- Se conclui essa mesma votação

§

Segurança.

Considere que K(X) significa "X cifrado/encriptado com a chave K". Considere que a seguinte mensagem genérica é enviada pela Alice para o Bob

$$K_1(H(M)), K_3(K_2), K_4(M)$$

sendo M a mensagem propriamente dita (payload), K_1 a K_4 chaves criptográficas, H uma função de hash e as vírgulas apenas separam as diferentes partes da mensagem. Em relação às chaves criptográficas:

- K_1, K_2, K_3 e K_4 não são necessariamente distintas (p.ex. pode acontecer que $K_i = K_j$ para algum par i, j);
- Algumas dessas chaves podem ser a chave pública ou a chave privada da Alice (respetivamente K_a^+ e K_a^-) ou do Bob (respetivamente K_b^+ e K_b^-).

Questão 67 O campo $K_1(H(M))$ tem por objetivo garantir a autenticidade e a integridade da mensagem, ou seja, que foi obrigatoriamente a Alice que enviou essa mensagem. Que tipo de cifra tem de ser usada: simétrica, assimétrica, outra?

Questão 68 Que chave deve usada como K_1?

Questão 69 Qual é a propriedade de segurança assegurada pelo campo $K_4(M)$?

Questão 70 Que tipo de cifra deve ser usada no campo $K_4(M)$: simétrica, assimétrica, outra? Justifique brevemente.

Questão 71 Que chave deve ser usada como K_3?

Questão 72 Algum par de chaves K_1, K_2, K_3, K_4 deve ser igual? Justifique brevemente.

Capítulo 1

Soluções

RPC	
Questão 1	*Vantagens:* Eficiência, menor overhead; Pode ser usado para comunicação 1 para n *Desvantagem:* Não garante fiabilidade na comunicação: pacotes podem ser perdidos, duplicados, reordenados.
Questão 2	Interface tipo mensagem.
Questão 3	Talvez.
Questão 4	1000
Questão 5	`increment(var)`: incrementa o valor da variável `var`
Questão 6	`increment(var)`: Web Services baseados em SOAP.
Questão 7	*Vantagem:* Inspeção de mensagens é mais simples devido ao formato textual *Desvantagem:* Formato menos eficiente
Questão 8	`increment(var)`: Permite desenvolver servidores apenas nas linguagens Java e Go.
Questão 9	DNS

Questão 10	`message requestIncrementar {`
	`  int32 incremento = 1;`
	`}`
	
	`message replyIncrementar {`
	`}`
	
	`message requestNumPessoas {`
	`}`
	
	`message replyNumPessoas {`
	`  int32 resultado = 1;`
	`}`
	
	`service praia {`
	`  rpc incrementar(requestIncrementar) returns (replyIncrementar);`
	`  rpc obter_num_pessoas(requestNumPessoas) returns (replyNumPessoas);`
	`}`

§

Sincronização de relógios físicos

Questão 11	1 ms
Questão 12	A com +20 segundos; B com -20 segundos; C com 0 segundos
Question 13	$R_{1.2}$: delta$_1$: $(220+8)-216=+12$, erro$_1$: ± 8 $R_{2.1}$: delta$_2$: $(204+7)-222=-11$, erro$_2$: ± 7
Question 14	soma dos erros de leitura: $\pm(8+7)=15$, diferença máxima = 15
Question 15	A segunda alternativa é melhor. A precisão é afectada apenas pelo erro de uma das leituras no primeiro algoritmo, cada leitura tem um erro e a diferença máxima é a soma dos dois erros

§

Relógios lógicos

Questão 16	A é concorrente com B					
Questão 17	[3,1]					
Questão 18	A	[101, 100, 100]	B	[100, 101, 100]	C	[100, 102, 100]
	D	[101, 100, 101]	E	[102, 100, 100]	F	[102, 102, 102]
Questão 19	B	101				
	C	102				
Questão 20	A	[101, 100, 100]	B	[100, 101, 100]	C	[102, 100, 100]
	D	[101, 100, 101]	E	[103, 100, 101]	F	[101, 101, 102]
Questão 21	B	101				
	C	102				
Questão 22	Com relógios lógicos, sabemos que C não pode acontecer-antes de B, mas nada nos diz se B aconteceu-antes de B. Com relógios vetoriais, sabemos que B e C são concorrentes.					

§

Exclusão mútua distribuída

Questão 23	r_1	envia ok a p_1
	r_2	coloca o pedido de p_3 pendente na fila
	r_3	envia ok a p_1
	r_4	envia ok a p_3
Questão 24	O identificador do processo é usado para desempatar	

Questão 25	process	p_1	p_2	p_3
	state	HELD	WANTED	RELEASED
	voted	TRUE	TRUE	TRUE
	pending	$\emptyset$	$\{p_2\}$	$\emptyset$

§

Estados globais	
Questão 26	O estado global S2é coerente (todas as mensagens recebidas foram também enviadas)
Questão 27	O estado global S3 é coerente (todas as mensagens recebidas foram também enviadas)
Questão 28	O estado global S2 é coerente (todas as mensagens recebidas foram também enviadas)
Questão 29	O estado global S3 é incoerente (a segunda mensagem de p_3 foi recebida mas não enviada)
Questão 30	Incoerente, S8 implica a presença de S9, mas este está em falta
Questão 31	Coerente. S12 implica a presença de S10 (OK). S8 implica a presença de S9 (OK). S9 implica a presença de S6 (OK).

§

Eleição de líder	
Questão 32	...só irá encaminhar mais uma mensagem de "election"
Questão 33	No melhor cenário, o algoritmo de "Bully" consegue terminar a eleição trocando menos mensagens que o de anel.
Questão 34	Enviou uma mensagem de "election"só para p_3 e p_4

§

Registos	
Questão 35	Figura 7(a): registo atómico Figura 7(b): registo unsafe Figura 7(c): registo safe Figura 7(d): registo regular
Questão 36	A Figura 8(a) ilustra uma execução que não é valida para registos "safe", "regular" ou "atomic" A Figura 8(b) ilustra uma execução que é válida para registo "atomic". A Figura 8(c) ilustra uma execução válida para registos "regular"
Questão 37	Atomic.
Questão 38	É preciso ler de uma maioria, escolher o valor mais recente, e escrever esse valor numa maioria.
Questão 39	10
Questão 40	10 ou 12, dependendo das respostas que cheguem primeiro ao cliente.

§

Espaços de tuplos	
Questão 41	Esta operação precisa de ser executa com ordem total em relação às outras operações de Take
Questão 42	Chama a primitiva "take"para eliminar este tuplo e depois chama a primitiva "put"para colocar o novo valor do tuplo.

§

Replicação fracamente coerente			
Questão 43	Servidor	Pode servir o pedido (sim/não)?	relógio retornado (caso retorne algum)
	S_1	No	-
	S_2	Não	-
	S_3	Sim	(2, 5, 8)
Questão 44	Servidor	Pode servir o pedido (sim/não)?	relógio retornado (caso retorne algum)
	S_1	Sim	(3, 5, 5)
	S_2	Sim	(2, 7, 5)
	S_3	Sim	(2, 5, 9)
Questão 45	As operações A e B		
Questão 46	A réplica não responde e não altera o seu relógio		

§

Replicação	
Questão 47	Salvaguarda-recuperação
Questão 48	Os pedidos necessitam de ser entregues por ordem total.
Questão 49	Propaga o seu novo estado aos secundários e espera que todos (exceto os secundários em falta) respondam com ack, para finalmente responder ao cliente.
Questão 50	Na replicação ativa as réplicas executam as operações de escrita (updates) depois de chegarem a acordo sobre a ordem dos updates.

§

Difusão totalmente ordenada	
Questão 51	4
Questão 52	Sim, mensagem B
Questão 53	6
Questão 54	O máximo dos números propostos pelos participantes

§

Sincronia na Vista	
Question 55	Execução 1, Execução 2
Question 56	Execução 3 (A e C entregues a p_3 mas não a p_1 nem a p_2)
Question 57	Execução 1 (B entregue antes da vista a p_1 e depois da vista a p_2) Execução 2 (C entregue antes da vista a p_2 e depois da vista a p_1)
Question 58	CLEANUP () { }

§

Consenso	
Questão 59	Qualquer um dos valores propostos
Questão 60	3
Questão 61	*condicao*: participantes $\subseteq$ processosactivos *parametro*: MAX(propostas) *instrucao*: cnmpDecide(V)

§

Transações distribuídas		
Questão 62	Este protocolo pode bloquear se o coordenador falhar	
Questão 63	Não. Por exemplo, se um dos participantes for mais rápido e executar confirmar na sua transacção local e depois um outro, um pouco mais lento, quiser abortar, já não há forma de voltar atrás, não se garantido atomicidade. Por isso, á necessário ter uma segunda fase para dar a oportunidade a qualquer participante de abortar unilateralmente.	
Questão 64	Decidir que o desfecho da transacção distribuída é ABORT.	
Questão 65	Têm obrigatoriamente de continuar à espera.	
Questão 66	É atribuído um identificador único à transacção: O serviço de voos se junta à transacção: Se inicia a votação do protocolo 2PC: Se conclui essa mesma votação:	2 4 7 7

§

Segurança	
Questão 67	Assimétrica
Questão 68	K_a^-
Questão 69	Confidencialidade da mensagem M
Questão 70	Simétrica por questões de desempenho
Questão 71	K_b^+
Questão 72	$K_2 = K_4$

§