

Mestrado Integrado em Engenharia Informática – Algoritmos e Sistemas Distribuídos

Primeiro teste – 20 de outubro de 2014 – Duração: 1 hora e 30 minutos

Responda às questões no espaço reservado para o efeito após cada alínea, ou na última página caso necessite mais espaço. Indique claramente qualquer pressuposto que tenha de fazer para resolver a uma questão.

Número	Nome
--------	------

1. [10 valores] O problema da eleição de um líder é formulado da seguinte forma: o problema não necessita de inputs, e o output é feito através de uma variável $leader_i$ cujo conteúdo pode mudar ao longo da execução (correspondendo a vários outputs) e que contém o identificador de um processo que o processo i decide eleger como sendo o líder actual. A única condição que correção diz que os processos devem, a partir de certo momento (“eventually”), todos eleger um mesmo líder j que é um processo correto. (Definimos um processo como sendo correto se não sofre uma falha por crash ao longo de toda a execução, sendo que consideramos um modelo de falhas por crash sem recuperação.) De forma mais precisa, a condição de correção é:

[C1] Existe um processo correto com o identificador j , e um tempo após o qual (ou seja, “eventually”) qualquer processo i que não sofreu uma falha por crash faz output de $leader_i = j$.

(a) Esta é uma condição de safety ou de liveness?

(b) Considere o seguinte trace em que nenhum dos processos ($p1$ e $p2$) tem falhas, e onde “ $pi.leader(pk)$ ” indica que o processo pi fez output da eleição de pk como o líder:

$p1.leader(p1)$, $p2.leader(p2)$, $p1.leader(p2)$, $p1.leader(p1)$

Este trace obedece a C1? Caso C1 seja uma propriedade de liveness e a resposta for negativa, extenda o trace de forma a obedecer a C1.

(c) Considere a seguinte possível solução para este problema: cada processo envia uma mensagem de “ping” para todos os outros processos e espera por um “ping-reply” (que é enviado automaticamente como resposta a um ping) até a um timeout T . O output $leader_i$ é determinado pelo processo com o identificador mais baixo dentro de todos os que responderam antes de T .

Esta solução resolve o problema da eleição de um líder sob o pressuposto de um sistema síncrono em que T é o tempo máximo para transmitir+processar mensagens? Se sim demonstre porquê, se não dê um contra-exemplo.

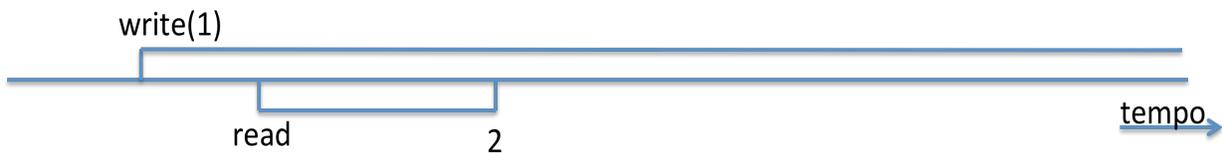
(d) A mesma solução resolve o problema sob o pressuposto de um sistema assíncrono? Se sim demonstre porquê, se não dê um contra-exemplo.

(e) A eleição de um líder pode ser usada para estender o Paxos de forma a que este passe a garantir a condição de liveness (desde que menos de metade dos processos falhem por crash), e preservando também as condições de safety. Por outras palavras: o Paxos, em conjunto com uma solução para a eleição de líder, passa a resolver o consenso num sistema assíncrono. Tendo em conta esta possibilidade, indique se o problema da eleição de um líder tem solução num sistema assíncrono em que pelo menos um dos processos possa falhar por crash? Demonstre a sua resposta.

2. [10 valores] Considere o problema de implementar um variável de leitura/escrita atômica. Suponha que valor inicial da variável é zero.

(a) Indique se os seguintes traces de execuções obedecem à atomicidade. Se sim, indique na figura possíveis pontos de serialização. Se não, justifique sucintamente porquê.

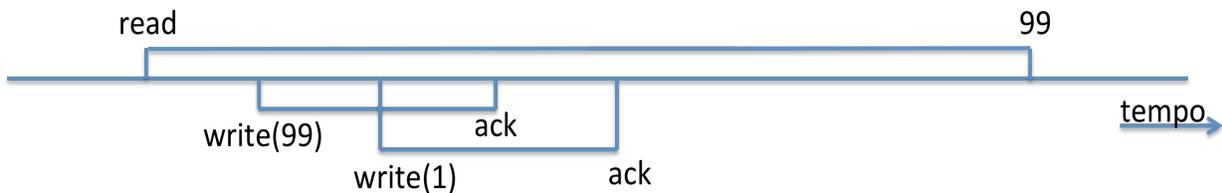
(a.1)



Atômico? _____

Justificação:

(a.2)



Atômico? _____

Justificação:

Número: _____

Nome: _____

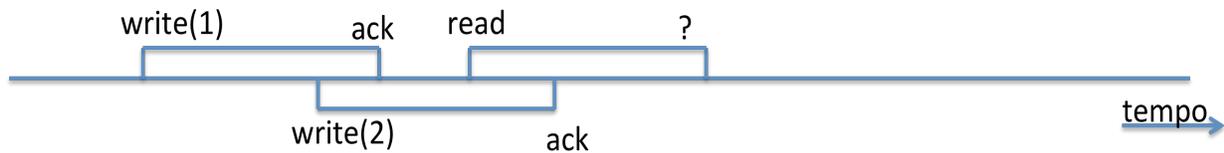
(b) Para os outputs indicados com '?', indique todos os outputs que seriam permitidos de forma a obedecer à condição de atomicidade.

(b.1)



Possíveis valores: _____

(b.2)



Possíveis valores: _____

(c) Considere a seguinte alteração ao algoritmo ABD para replicar variáveis de leitura/escrita: suponha que se retira a primeira fase da operação de escrita (a fase de read-tag), e ao invés a "tag" usada na segunda fase que agora passa a ser a única (a fase de write) é obtida através de um contador que cada processo cliente incrementa por cada escrita, contador esse que é concatenado com o identificador do processo de forma a clientes diferentes usarem "tags" diferentes.

c.1) Mostre, através de um diagrama temporal de uma execução do algoritmo modificado, que esta modificação não garante a atomicidade.

c.2) Indique uma restrição (tão pequena quanto possível) ao número e ao comportamento dos processos clientes (em termos do tipo de operações que estes podem invocar) de forma a que, com essa restrição, o algoritmo passe a garantir a atomicidade.

Espaço adicional para resolução do teste. Indique junto à pergunta caso tenha de usar este espaço.