

Sistemas Distribuídos*Exame de Recurso*

9 de Fevereiro de 2015

Duração: 2h00m

I

- 1 Explique como funcionam genericamente as operações de `lock` e `unlock` ao nível do sistema operativo.

As operações `lock` e `unlock` disponibilizadas pelo sistema operativo eliminam as esperas ativas no acesso a regiões críticas. Um processo que se encontre na secção crítica, impede os outros que pretendem entrar, de obter o `lock`, passando ao estado de bloqueado, não sendo escalonados e não consumindo tempo de processador. No fim de execução de região crítica, o processo invoca o `unlock`, libertando o `lock`, que muda o estado dos outros processos de bloqueado para pronto, podendo estes de seguida, serem escalonados.

- 2 Indique e descreva sucintamente as arquitecturas descentralizadas de sistemas distribuídos que estudou.

As arquitecturas descentralizadas dividem-se em dois tipos, estruturadas e não estruturada.

As estruturadas possuem uma estrutura organizacional bem definida.

Nesta, a totalidade do domínio dos dados, é dividida pelo contradomínio dos constituintes da arquitetura através de uma função de hash. ex : DHT em p2p.

As não estruturadas não possuem qualquer tipo de estrutura, tratam-se de um conjunto de nós dispersos, cada um possuindo uma determinada vista parcial da rede. Estas baseiam-se no princípio básico de aleatoriedade, fator responsável pela troca de modos das vistas de qualquer dos nós constituintes. ex: Peer to peer através de métodos aleatórios..

- 3 Discuta as diferenças ao assegurar exclusão mútua entre processos a executar numa única máquina e num sistema distribuído.

A exclusão mútua processa-se de forma diferente caso haja realizado numa máquina ou num sistema distribuído.

Numa máquina é alcançado através das primitivas de sincronização `lock` e `unlock` disponibilizados pelo sistema operativo, que permitem bloquear processos enquanto não conseguirem obter o `lock` de acesso à região crítica.

Em sistemas distribuídos, quer centralizados ou descentralizados este é obtido por trocas de mensagens. O cliente envia uma mensagem a pedir permissão para escutar a região crítica e só poderá executar se o coordenador, no caso das centralizadas, ou todos os outros constituintes, no caso das descentralizadas, enviarem uma mensagem de resposta

II

Considere um sistema cliente/servidor de uma variante do jogo de "Batalha Naval". O servidor faz a gestão de uma grande matriz de $n \times m$ de submarinos; cada posição da matriz vale 1 se estiver ocupada ou 0 se estiver livre. Suponha que no início de cada sessão o conjunto de jogadores é conhecido e a matriz é inicializada de forma aleatória por uma função `iniciar`, já existente. O servidor aceita pedidos do tipo `disparo` cujos argumentos identificam o cliente e as coordenadas do disparo (por exemplo, "joao 13 245"). A pontuação de cada disparo é calculada somando todos os submarinos da linha e da coluna que são afundados pelo disparo. Uma vez afundado, um submarino já não volta a contar para outro disparo.

Cada jogador pode efectuar 3 disparos. No terceiro disparo, o cliente é bloqueado até ao final do jogo, momento em que recebe como resposta o identificador do jogador vencedor. O jogo termina quando todos os jogadores usarem os seus disparos.

Escreva em Java o código do servidor de forma a que atenda eficientemente pedidos concorrentes.

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or other markings on the paper.