

Sistemas DistribuídosExame¹

26 de julho de 2019

Duração: 2h00m

Por favor responda a cada um dos 2 grupos em folhas de teste separadas. Obrigado.**I****1 Distinga comunicação *síncrona* de *assíncrona* em sistemas distribuídos. Dê exemplos de *middleware* para cada uma delas.**

Comunicação Síncrona : O emissor e o recetor das mensagens estão sincronizados, ou seja, +para haver comunicação o cliente e o servidor têm de estar sincronizados. O envio e a recepção de mensagens são ambas ações bloqueantes, isto é, quando um cliente/emissor emite uma mensagem este fica bloqueado até obter uma resposta e enquanto o recetor/servidor não recebe uma mensagem este fica também bloqueado.

Comunicação Assíncrona : Neste caso, o envio não é uma ação bloqueante, ou seja, o emissor pode enviar uma mensagem e continuar a sua execução logo após o envio da mesma, isto porque a mensagem é copiada para um buffer/queue de mensagens. A transmissão destas mensagens ocorre em paralelo com a execução do emissor. No caso da recepção de mensagens, a recepção tanto pode ser uma ação bloqueante ou não.

Exemplos de Middleware para a comunicação síncrona são: Message passing em MOM (Middleware Orientado a Mensagens)

Exemplos de Middleware para a comunicação assíncrona são: Message Queuing em MOM (Middleware Orientado a Mensagens)

2 Qual a razão para estruturar uma aplicação distribuída em camadas? Use um exemplo.

Modelo em três camadas (3-Tier), derivado do modelo 'n' camadas, recebe esta denominação quando um sistema cliente-servidor é desenvolvido retirando-se a camada de negócio do lado do cliente. O desenvolvimento é mais demorado no início comparando-se ao modelo em duas camadas porque é necessário dar suporte a uma maior quantidade de plataformas e ambientes diferentes. Em contrapartida, o retorno vem em forma de respostas mais rápidas nas requisições, tanto em sistemas que rodam na Internet ou em intranet, e mais controle no crescimento do sistema.

As três partes de um ambiente modelo três camadas são: camada de apresentação(Interface), camada de negócio e camada de dados. Características esperadas em uma arquitetura cliente-servidor 3 camadas:

- O software executado em cada camada pode ser substituído sem prejuízo para o sistema;
- Atualizações e correções de defeitos podem ser feitas sem prejudicar as demais camadas. Por exemplo: alterações de interface podem ser realizadas sem o comprometimento das informações contidas no banco de dados.

A separação em camadas lógicas torna os sistemas mais flexíveis, permitindo que as partes possam ser alteradas de forma independente. As funcionalidades da camada de negócio podem ser divididas em classes e essas classes podem ser agrupadas em pacotes ou componentes, reduzindo as dependências entre as classes e pacotes; podem ser reutilizadas por diferentes partes do aplicativo e até por aplicativos diferentes. O modelo de 3 camadas tornou-se a arquitetura padrão para sistemas corporativos com base na Web.

3 Explique como funciona um protocolo de exclusão mútua distribuída centralizado. Identifique as suas principais vantagens e desvantagens.

O Protocolo de exclusão mútua distribuída tem base na assumpção de que o sistema consiste de n processos e cada processo está no seu processador. Cada processo tem uma zona crítica que requer exclusão mutua. Tem como requisito que se um processo se encontra a executar na zona crítica, então mais nenhum processo se encontra a executar a sua. O algoritmo centralizado deste protocolo consiste num processo que é escolhido para coordenar o acesso à zona crítica. Qualquer processo que queira executar a sua zona crítica tem de enviar um pedido ao processo coordenador. O coordenador decide se o processo pode entrar na zona crítica e envia uma resposta. Quando o processo recebe a respostas do coordenador, o processo inicia a execução da zona crítica. Quando este acaba a execução da mesma, o processo envia uma mensagem ao coordenador a libertar a zona.

Desvantagens: Se o coordenador falhar o sistema cai.

Os processo que ficam bloqueados quando enviam um pedido não conseguem distinguir um coordenador que "n" de uma "acesso negado" pois neste caso nenhuma mensagem é devolvida.

Num sistema de maiores dimensões um coordenador pode se tornar numa performance bottleneck.

Vantagens: Algoritmo justo. Os processos executam a zona crítica por ordem de chegada.

Garante que apenas um processo entra na zona crítica simultaneamente.

Não permite starvation, ou seja, nenhum processo espera para sempre.

Simple e facil de implementar.

Requiere apenas 3 mensagens por uso de recurso.

¹Cotação — 10+10