

Sistemas Distribuídos

(exame da época normal, duração 1h30m)

02.02.2022

responda às perguntas de forma sucinta e sempre justificando

1. Qual a diferença entre a *Software Architecture* e a *System Architecture* de um sistema distribuído? Como se designam os níveis em cada uma destas descrições de uma arquitetura?
2. Descreva duas vantagens da utilização de vários *threads* em clientes numa aplicação distribuída. Descreva duas vantagens da utilização de vários *threads* em servidores numa aplicação distribuída.
3. Qual é a diferença entre comunicação síncrona e assíncrona? Qual é a diferença entre comunicação transiente e persistente? Dê um exemplo de uma aplicação distribuída baseada em comunicação síncrona e transiente.
4. O que é um *Message-Oriented Middleware* (MOM)? Que tipo de acoplamento espacial e temporal tem um tal sistema? Dê um exemplo de um cenário de aplicação onde seria natural a sua utilização.
5. Em que consiste o mecanismo designado por *Remote Procedure Call* (RPC)? Descreva sucintamente como é implementado e as suas vantagens relativamente à comunicação baseada em *sockets*.
6. Em que consiste a comunicação isócrona? Indique duas técnicas usadas para minimizar os efeitos da latência e perda de pacotes em sistemas que implementam este tipo de comunicação. O que mede a métrica de qualidade de serviço designada por *jitter*?
7. Descreva o algoritmo *Anti-Entropy* de disseminação de informação numa rede *peer-to-peer*? Que operações básicas implementa? Qual é a complexidade de divulgar uma atualização para todos os nós em uma rede?
8. O serviço de resolução de nomes numa rede de computadores mapeia nomes de máquinas em endereços IP. No caso de máquinas na Internet, como se inicia o processo de resolução? Qual a importância da utilização de *caches* neste processo?
9. Em que consiste uma *Distributed Hash Table* (DHT) e para que poderia ser utilizada, por exemplo, numa rede *peer-to-peer*, com vantagens em termos de performance.
10. Em que consiste o modelo de consistência sequencial quando consideramos um cenário em que vários clientes realizam operações de leitura/escrita sobre um *data store* constituído por várias réplicas da mesma informação, e.g., uma base de dados? Que algoritmo conhece que poderia ser utilizado para implementar este modelo de consistência?