

Sistemas Distribuídos

1ª Lista de Exercícios

Resolução: Em grupo de 2/3 pessoas (mesmo grupo dos trabalhos anteriores)

Entrega: 26/09 (sexta-feira-feira)

1. O que significa dizer que um Sistema Distribuído é transparente? Dê exemplos de diferentes tipos de transparências.
2. Qual(is) é (são) a(s) motivação(ões) para projetar Sistemas Distribuídos Abertos?
3. Cite características que possam determinar a diferença entre um Sistema Operacional de Rede e um Sistema Operacional Distribuído.
4. Quais características permitem definir uma arquitetura cliente/servidor de três camadas?
5. Qual é a diferença entre distribuição vertical e distribuição horizontal?
6. Ao invés de permitir que um servidor registre-se em um *daemon* (como no caso de DCE-RPC), podemos determinar que uma aplicação sempre utilize a mesma porta. Essa porta pode ser utilizada em referências para objetos no servidor. Quais são as desvantagens desse esquema?
7. Considere a operação de leitura de um arquivo em um servidor *single-threaded* e em um servidor *multithreaded*. O tempo para receber a requisição e realizar o processamento é de 15 msec mais 75msec (durante esse tempo o *thread* “dorme”) de acesso ao disco. Quantas requisições/segundo um servidor *single-threaded* pode manipular? Quantas requisições/segundo serão manipuladas se o servidor for *multithreaded*?
8. Compare as arquiteturas de *thread worker pool* e *thread-per-request*.
9. Comente sobre vantagens/desvantagens de limitar o número de *threads* de um servidor.
10. Mencione situações onde podem existir vantagens/desvantagens em implementar servidores baseados em processos concorrentes em contraposição aos servidores *multithreaded*.
11. Considere um processo *P* que requer acesso para um arquivo *F*. Esse arquivo está localizado na mesma máquina onde *P* está executando. Quando *P* é

movido para uma máquina deve continuar acessando F. Como isso pode ser conseguido?

12. Suponha que dois processos detectam simultaneamente a perda de um coordenador e ambos decidem forçar uma eleição usando o algoritmo de *Bully*. O que acontece nesse caso?
13. O Algoritmo de Ricart e Agrawala tem um problema que se um processo “caído” e nenhuma resposta é enviada para o processo que pediu acesso à Região Crítica, a falta de resposta será interpretada como uma “negação” de serviço. Alguns autores sugerem que todas as requisições sejam respondidas imediatamente com o intuito de detectar situações de *crash*. Há circunstâncias onde mesmo esse método pode falhar? Em caso positivo, comente sobre essas circunstâncias.
14. É possível sincronizar o *clock* de dois computadores ligados por uma rede local sem uma referência a uma fonte externa de tempo? Quais fatores limitam a solução que você sugeriu?