

SORULAR

1. Define what is meant by a distributed system? What are the characteristics of a distributed system?
2. What is the role of middleware in a distributed system?
3. Explain what is meant by (distribution) transparency, and give examples of different types of transparency.
4. Why is it not always a good idea to aim at implementing the highest degree of transparency possible? Give an example for such cases.
5. Scalability can be achieved by applying different techniques. What are these techniques?
6. Explain what is meant by a virtual organization and give a hint on how such organizations could be implemented.
7. If a client and a server are placed far apart, we may see network latency dominating overall performance. How can we tackle this problem?
8. What is three-tiered client-server architecture?
9. What is the difference between a vertical distribution and a horizontal distribution?
10. In a structured overlay network, messages are routed according to the topology of the overlay. What is an important disadvantage of this approach?
11. Would it make sense to limit the number of threads in a server process? Explain.
12. Statically associating only a single thread with a lightweight process is not such a good idea. Why not?
13. Having only a single lightweight process per process is also not such a good idea. Why not?
14. Presume that a process P on a computer D wants to contact a process Q on another computer E. Give different principle for how this can be accomplished?
15. Describe what is meant by Stateless vs. Stateful Servers when using Client-Server style. How do you choose between them and what will be the corresponding consequences?
16. Discuss peer-to-peer architectural model for construction of distributed systems.
17. Compare threads with processes. Give at least one similarity and one difference?
18. What is the difference between a distributed operating system and a network operating system?
19. List three disadvantages and three advantages of distributed systems.
20. Give three examples of scalability limitations. Explain one of them.
21. How does context switching occur a) between processes and b) between threads? Which one is more efficient? Why?
22. Do you get (or emulate) parallelism with a single thread? If yes, How? If no, Why?
23. What are the advantages and disadvantages of user level threads?
24. How do you combine the advantages of user and kernel level threads with lightweight processes (LWP)?
25. At any specific time, how many threads can be running in the system? Is the question correct? Correct the question and answer it?
26. What are the sender-initiated and receiver-initiated code migrations? Give examples.