(**V.0.3**, 29/November/2017)

Courses on Computer Networks and Distributed Systems

Marshaling and Remote Invocation

For working the following questions, **first, download and study the pdf slides from paloalto** and the corresponding textbook sections.

1. Is the Request-Reply (RR) a standardized protocol like IP, for example?
2. This question is related to the RR protocol implemented over TCP, why do we say that TCP, in this case, is redundant?
3. TCP imposes an excessive overhead over the RR protocol, can you explain why?
4. When the RR protocol is implemented over UDP, can any delivery guarantees be provided?
5. In an implementation of the RR protocol over UDP, what resource is used by a client to check that a response is not a repeated response from the server?
6. In the RR protocol, if a request message is retransmitted, the server may receive it many times, does this cause the server to execute an operation more than once?
7. What is a persistent connection in HTTP? All versions of this protocol support persistent connections?
8. What are the MIME extensions used for in the context of http and html?
9. Is RPC the same as RMI? Explain the most important differences.
10. What are the most important transparencies in RPC and RMI?
11. Investigate what the Unix RPC and Java RMI call semantics are.
12. In the context of RPC and RMI, what is an *idempotent operation?*
13. In page 206 of DK the authors state "*Distributed object systems **may** adopt the client-server architecture".* Explain this statement.

14. Is true that having client and server objects in different processes enforces encapsulation? What is the meaning of encapsulation in this case?

15. What is a *remote interface*? And a *remote object reference?*

16. What is Java RMI call semantics?

17. In Java RMI, what role plays the *Dynamic Class Loading?* Are *stub classes* dynamically loaded?

18. In a Java RMI distributed object system who provides a remote object's stub instance to a client? And, who provides the stub's class file? What does the client need the stub's class? Who provides the stub's class, normally?

19. The Java serialization of remote object references is the same as that of common objects? Explain what additional information is included in a remote object reference serialization.

20. Is Java RMI multithreaded?

21. Did you successfully complete the RMI practical?

22. Try to create a diagram of the remote invocation that resembles the '*live explanation'* I provided you in the lab (You can download it from paloalto.unileon.es/asd), recall the following points:

    a. The Server instantiates a remote object (the servant)
    b. The server registers the servant with the Java registry running on the server's host
    c. When the client contacts the server for a remote invocation, the server provides it with a stub class that represents the servant in the client
        i. The stub class is transferred to the client
        ii. In the server, a skeleton class is created which plays an equivalent role to that of the stub, but in the server