

Excercise Sheet 3

(To be processed until 15.05.2020)

Lecture Distributed Systems Summer Term 2020

Exercise 1: RPC and unions

The programming language C supports a concept called union. A field of a data record can contain one of several alternatives, whereby the data in the field can have a different type for each alternative. However, there is no really safe method of checking at runtime which variants are contained in a field, that is, which data type the field currently has. Do unions have any effect on remote procedure calls, and if so, which ones?

Exercise 2: Remote object references

Give an example implementation for a remote object reference, that is, an object that contains all the required data for a client to interact with a remote object.

Exercise 3: Exceptions

Java and other programming languages support exceptions that are thrown when an error occurs. Explain how you would implement exceptions in RPCs and RMIs and what special features need to be considered.

Exercise 4: Threads

A client makes remote procedure calls to a server. The client takes five milliseconds to calculate the arguments for a request, and the server takes ten milliseconds to process a request. The local operating system processing time for each send or receive operation is 0.5 milliseconds and the network time for transmitting the request or response messages is three milliseconds. Marshalling and unmarshalling require 0.5 milliseconds per message. Calculate the time it takes for the client to generate two requests and to receive the corresponding replies

- a) if it is single-threaded, and
- b) if it has two threads that can execute the requests concurrently on a single processor.

You can ignore times for context switching.

Exercise 5: RMI basics and architecture

With Java *Remote Method Invocation* (RMI) it is possible to call and use methods of remote objects. RMI is based on RPC (Remote Procedure Call) and a three-layer architecture.

- a) Name the differences between RMI and RPC.
- b) List the three different layers of RMI and briefly describe their functionality.

- c) Briefly describe what a proxy object is.
- d) What does reflection mean in this context?

Exercise 6: Programming: Java RMI - Hello-World!

In the lecture you were instructed how to implement a first Hello-World program with Java RMI. Now reproduce the necessary steps yourself. Implement

- a) the interface for the remote object,
- b) the implementation class for the remote object,
- c) a server application to host the remote object,
- d) a client application with calls to the remote object.

Finally, compile and start the system. To start the naming service, run `rmiregistry` on the default port 1099, then start the server and client in two console windows.